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

[UCI]I804:24011-200000266968
조선대학교
CHOSUN UNIVERSITY

2018년 8월

석사학위 논문

유의어 가중치를 적용한 논문 표절 유사도 측정 방법

조선대학교 산업기술융합대학원

소프트웨어융합공학과

김 민 범

조선대학교
CHOSUN UNIVERSITY

2018년 8월

석사학위논문

유의어 가중치를 적용한 논문 표절 유사도 측정 방법

김 민 범

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

유의어 가중치를 적용한 논문 표절 유사도 측정 방법

Method of measuring paper plagiarism similarity using synonym weighting

2018년 8월 24일

조선대학교 산업기술융합대학원

소프트웨어융합공학과

김 민 범

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

유의어 가중치를 적용한 논문 표절 유사도 측정 방법

지도교수 신 주 현

이 논문을 공학석사학위신청 논문으로 제출함.

2017년 4월

조선대학교 산업기술융합대학원

소프트웨어융합공학과

김 민 범

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

김민범의 석사학위논문을 인준함

위원장 조선대학교 교수 김 판 구 (인)

위 원 조선대학교 교수 반 성 범 (인)

위 원 조선대학교 교수 신 주 현 (인)

2018년 5월

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. 관련 연구 ····· 4
A. TF-IDF ····· 4
B. Word2Vec 와 Doc2Vec ····· 5
1. Word2Vec ····· 5
2. Doc2Vec ····· 6
C. DBSCAN ····· 8
D. 유사도 측정 ····· 10

Ⅲ. 유의어 가중치를 적용한 유사도 측정 방법 ····· 11
A. 시스템 구성도 ····· 11
B. 유의어 추출 ····· 13
1. 전처리 과정 ····· 13
2. Doc2Vec ····· 14
3. DBSCAN 군집화 ····· 15
4. 단락별 가중치 ····· 16
C. L-R 구문분석 ····· 20

Ⅳ. 실험 및 결과 ····· 22
A. 유사도 측정 ····· 22
B. 유사도 평가 ····· 26

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅴ. 결론 및 향후연구 ····· 28

참고문헌 ····· 30

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 목 차

[표 1-1] 교육인적자원부 논문 표절 가이드라인(2008.02) ····· 1
[표 1-2] 카피킬러 표절 검사 기준 ····· 2
[표 3-1] 품사 테그셋 ····· 13
[표 3-3] 문서 A의 각 단락별 거리 값 예 ····· 19
[표 3-4] R 데이터 예 ····· 20
[표 4-1] 실험 환경 ····· 22
[표 4-3] 기준값 측정 ····· 23
[표 4-3] 문장별 유사도 비교 ····· 25
[표 4-4] 유사도 평가 ····· 26
[표 4-5] 유사도 검출 문장 예 ····· 27

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그 림 목 차

[그림 2-1] CBOW(좌) 과 Skip-gram(우) 모델 ····· 5
[그림 2-1] Distributed Memory Model(DM) ····· 6
[그림 2-2] Distributed Bag of Words(DBOW) ····· 6
[그림 2-3] DBSCAN 클러스터링 ····· 8
[그림 3-1] 시스템 구성도 ····· 11
[그림 3-1] 단어 벡터값 ····· 14
[그림 3-2] 벡터값 .csv 파일 저장 ····· 14
[그림 3-3] DBSCAN 군집화 (단락별 가중치 적용하지 않음) ····· 15
[그림 3-4] DBSCAN 군집화 (단락별 가중치 적용) ····· 17

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

ABSTRACT

Method of measuring paper plagiarism similarity using synonym weighting

Minbeom Kim
Advisor : Prof. JuHyun Shin, Ph.D.
Department of Software Convergence Engineering
Graduate School of Industry Technology Convergence, Chosun University

With the advancement of the Internet, various information can be easily found without the restrictions of time and place. However, there are cases where such information is abused in plagiarizing others' theses. Plagiarism mostly occurs when using the content of other people's paper without citation. Some of the common forms of plagiarisms are plagiarizing text, ideas, rewriting other people's work, and interweaving various sources together in work. CopyKiller, the free service commonly used in plagiarism has standard processing criteria of judging plagiarism when six word-phrases are the same. However, this criterion seems insufficient since there can be rewordings.

The result of the earlier study showed that one document had an algorithm word A often appeared in the introduction and related studies but another word B which is a modification from the algorithm A, appeared in the main body. Since the two words are measured to be similar from the test, TF-IDF is modified to give weight by the paragraph of the introduction, related studies, main body, and conclusion, to improve the accuracy of similarity test on the

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

words alike but have different meanings.

Hence, this paper proposes a method of measuring similarity by finding synonyms between the words and applying weight by the paragraph for the improvement of accuracy in document similarity checks. After preprocessing the data, vector values of the word are extracted by using Doc2Vec and word with weights by paragraph is clustered by DBSCAN(Density-based spatial clustering of applications with noise). The main word in each cluster is set as the representative word, and other words in the same cluster are set as synonyms. The distance value between the representative word and a synonym is used as synonym weight.

Since the number is too low to be applied as weight with the distance value of the word alone, an L-R table is created via L-R syntactic parsing. The noun score is calculated using the score for the possible noun and is applied to the synonym weight.

To evaluate the similarity between the documents, similarity criteria by sentence are found first and the sentences having the similarity of 0.7 and higher are identified as the plagiarized sentence. The performance of the method suggested by this paper will be verified through a comparative evaluation with the existing studies.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅰ. 서론

A. 연구 배경 및 목적

인터넷의 발달로 인하여 시간과 장소에 제약되지 않고 다양한 정보를 쉽게 얻을 수 있다. 하지만 이러한 정보를 악용하여 소설, 드라마 등 다양한 표절의혹이 제기되고 있으며 논문 또한 여러 표절 사례가 존재한다[1].

일반적으로 표절은 타인의 논문의 내용을 출처를 표시하지 않고 사용하는 경우에 발생한다[2]. 표절의 대표적인 유형으로는 텍스트 표절, 아이디어 표절, 말 바꿔 쓰기 표절, 모자이크 표절 등이 있다. 텍스트 표절은 타인의 논문에서 글, 표, 그림, 그래프 등을 출처를 밝히지 않고 자신의 것처럼 사용하는 것이다. 아이디어 표절은 타인의 이론, 데이터, 방법 등의 독창적 아이디어를 출처를 밝히지 않고 자신의 생각으로 설명한 경우이다. 말 바꿔 쓰기 표절도 타인의 논문에서 내용을 자신의 방식으로 바꿔 쓰면서 출처를 표시하지 않는 경우이다. 모자이크 표절은 타인의 글을 활용하는데 있어 출처를 밝히지 않고 문장 변형, 편집 등을 하여 자신의 것처럼 만드는 것을 말한다[3].

다음 표[1-1]은 2008년 2월 교육인적자원부의 논문 표절 가이드라인이다[4].

[표 1-1] 교육인적자원부 논문 표절 가이드라인(2008.02)

표절 판정 기준
1. 여섯 단어 이상의 연쇄 표현이 일치하는 경우
2. 생각의 단위가 되는 명제 또는 데이터가 동일하거나 본질적으로 유사한 경우
3. 타인의 창작물을 자신의 것처럼 이용하는 경우
4. 남의 표현이나 아이디어를 출처표시 없이 쓰거나 창작성이 인정되지 않는 짜집기
5. 연구결과 조작 및 저작권 침해 가능성이 높은 저작물

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

논문 표절 검사에 많이 사용되는 Copykiller 무료 서비스의 표준 처리 기준을 보면 6어절 이상 동일한 경우 표절로 처리하는 방법을 적용 중이다.

[표 1-2] 카피킬러 표절 검사 기준

구 분	카피킬러 무료서비스
표절처리기준	카피킬러 표절처리기준 6어절이상 동일, 인용/출처포함, 목차/참고문헌 제외
검사문서	1일 3개 제한, 검사문서 1개당 1M, 3만자 제한
검사설정기능	없음
검사결과확인서	기본 형태의 확인서 제공(내 문서에 표절의심영역 체크)
검사결과확인	문장별 선택 보기

하지만 단어를 바꿔 쓰는 것에 있어 6어절 이상이 동일한 경우만을 표절로 판단하기에는 부족함이 있다. 본 논문에서는 단어를 바꿔 쓰는 경우에도 유사한 문서 검사의 정확성 향상을 위해 단어들 간 유의어를 찾아 문서 유사도를 측정하는 방법을 제안한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. 연구 내용 및 구성

본 논문에서는 텍스트 마이닝과 TF-IDF 관련 논문을 수집하여 불용어를 제거하고 Python 기반의 Doc2Vec을 사용하여 단어의 벡터값을 추출하고 DBSCAN 군집화와 L-R 구문분석을 통해 L-R Table을 구축하여 유의어를 찾고 서론, 관련 연구, 본론, 결론 마다 단락별 변형된 TF-IDF 가중치를 적용하여 문서 유사도를 측정하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 TF-IDF와 Doc2Vec 및 DBSCAN에 대하여 설명하고, 3장에서는 본 논문에서 제안하는 방법을 설명한다. 4장에서는 제안한 방법을 적용해서 문장의 유사도를 판별한다. 5장에서는 실험 결과와 향후 연구 방향을 기술한다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

Ⅱ. 관련 연구

A. TF-IDF

TF-IDF는 문서 간 단어 가중치를 측정하는 방법으로 단어를 포함한 문서의 빈도를 고려해 가중치를 측정하는 방법이다[5]. 식 (1) 과 같이 단어 빈도(Term Frequency) 와 역 문서 빈도(Inverse Document Frequency)의 곱으로 나타낸다.

$$W_{(t,d)} = tf_{(t,d)} \times idf_{(t)} \tag{1}$$

t는 단어, d는 문서이다. $tf_{(t,d)}$는 문서 d 내의 단어 t의 가중치를 의미한다. $idf_{(t,d)}$ 는 단어 t가 출현하는 문서 빈도에 역을 취하고 단어 t가 출현빈도가 높을수록 많이 쓰이는 단어이므로 가중치 값이 적게 적용된다. 역 문서 빈도는 다음 식 (2) 와 같다.

$$idf_{(t)} = \log(\frac{N}{df_{(t)}}) \tag{2}$$

N은 전체 문서 개수이다. $idf_{(t)}$는 로그를 취하여 값이 적게 나오므로 TF-IDF 값은 $tf_{(t,d)}$의 영향을 많이 받는다[6].

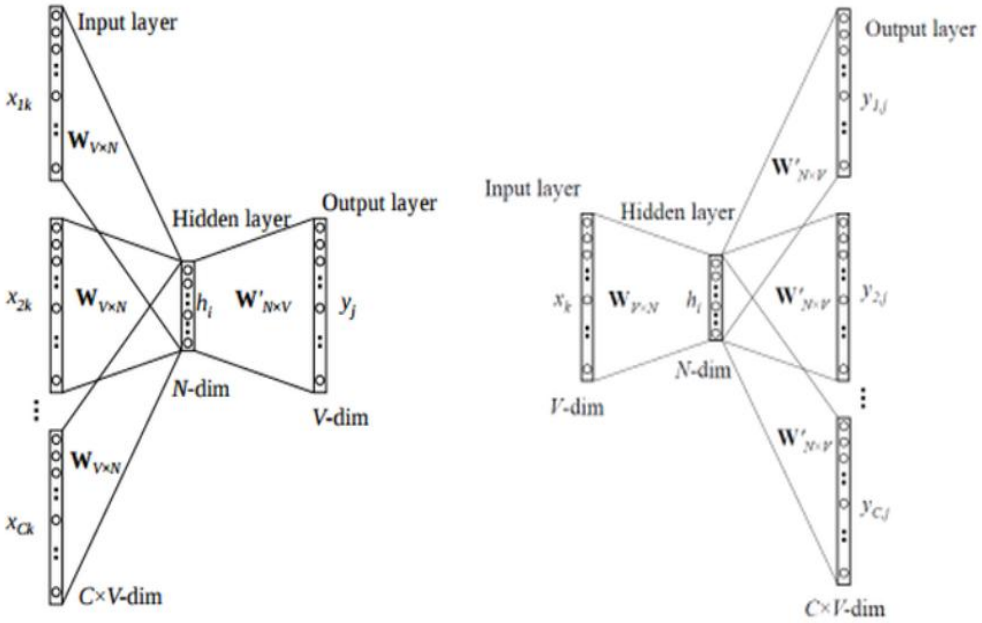
Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. Word2Vec 와 Doc2Vec

1. Word2Vec

Word2Vec 은 단어를 벡터로 표현하는 방법론이다. 유사한 의미의 어휘는 문맥에서 등장한다는 분산적 가정을 기반으로 단어를 연속적인 벡터 공간으로 임베딩한다[7][8]. 다음 [그림 2-1]은 각각 Word2Vec 모델인 Continuous Bag-of-Words(CBOW), Skip-gram 모델을 나타낸다.



[그림 3-1] CBOW(왼) 과 Skip-gram 모델

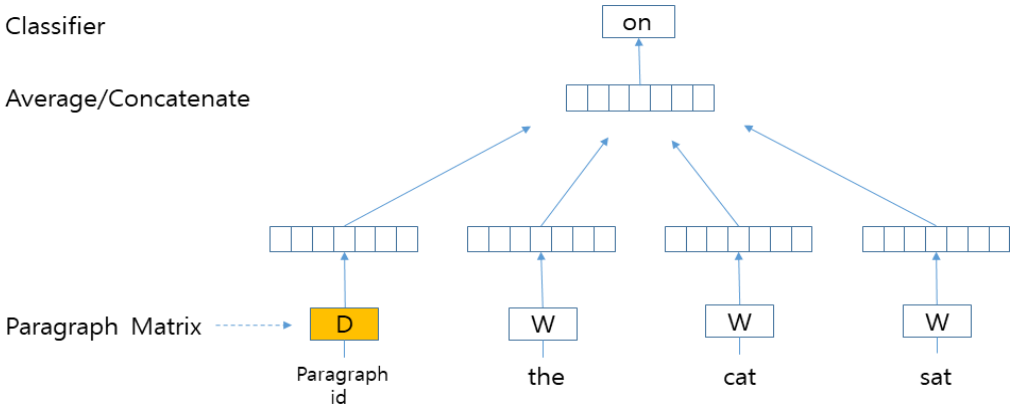
CBOW 모델의 경우 주어진 단어의 앞뒤의 단어를 Input으로 사용하여 주어진 단어를 맞추는 모델이다. Skip-gram 모델은 하나의 단어를 Input으로 사용하여 주위에 등장하는 나머지 단어들의 등장 여부를 찾는 모델이다[9].

Collection @ chosun

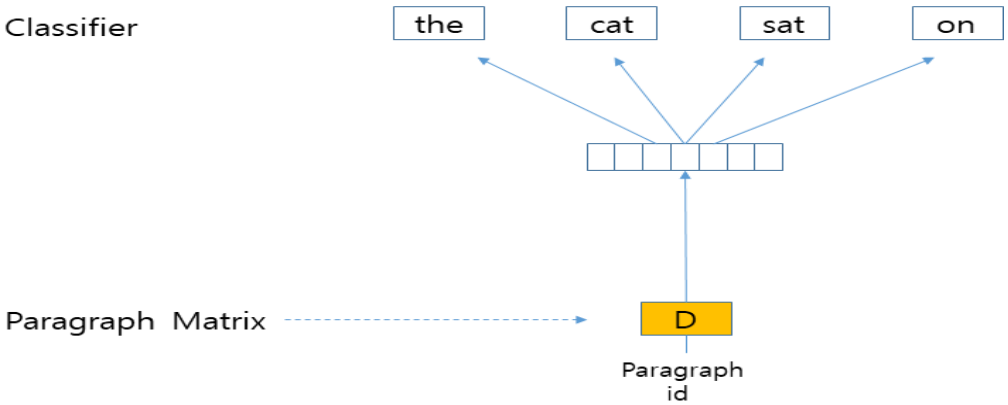
조선대학교
CHOSUN UNIVERSITY

2. Doc2Vec

Doc2Vec는 가변 길이의 텍스트를 고정 길이의 벡터로 표현하는 비지도 학습 알고리즘이다. Doc2Vec에는 다음 [그림 2-1]과 같이 distributed memory model(DM), [그림 2-2]와 같이 distributed bag of words(DBOW) 두 모델이 있다 [10].



[그림 2-1] Distributed Memory Model(DM)



[그림 2-2] Distributed Bag of Words(DBOW)

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

DM 모델은 벡터와 주변 단어 벡터를 이용하여 다음 단어를 예측할 수 있고, DBOW는 문서의 벡터로 문맥을 예측할 수 있다[11].

Doc2Vec는 Word2Vec 알고리즘과 같은 원리로 학습하고 문단 벡터(paragraph vecter)가 추가된 모델이다. 문단 내에 단어를 5개씩 순서대로 읽어 들여 다음 단어를 예측하는 과정으로 문단 벡터가 학습하게 된다.

김도우의 연구에서는 CNN(Convolutional Neural Network) 모델을 기반으로 Doc2Vec을 적용하여 문서 분류 성능을 향상시키는 연구를 진행하였다. Doc2Vec가 동일한 범주에 속한 문서에 대해 유사한 문서 벡터 표현을 생성하고 Word2Vec을 이용하여 생성한 단어 벡터 표현들을 CNN에 적용하였다[12].

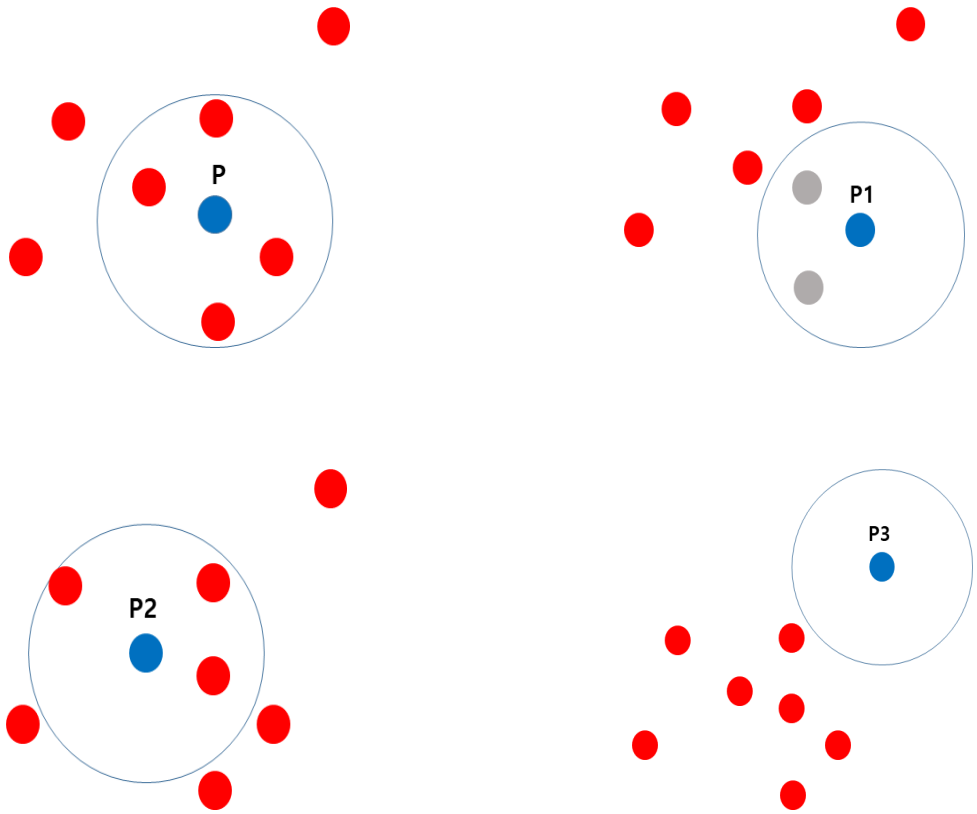
Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

C. DBSCAN

DBSAN은 Density model 중 하나이고 K-means와 같이 데이터의 위치 정보를 이용한다. K-means은 군집들 간 거리를 이용하여 클러스터링 하는 방법인데 DBSCAN은 밀도 기반의 클러스터링 방법으로 클러스터링 점이 세밀하게 밀집되어있어도 밀도가 높은 부분을 클러스터링 한다[13].

밀도가 가장 높은 핵심 벡터의 최소 기준 거리 내에 있고 인접 벡터의 개수가 일정 기준 이상일 때 하나의 클러스터를 형성한다. 이때, 핵심 벡터도 아니고 외곽 벡터도 아닌 벡터가 생성되는데 이 벡터를 노이즈라고 한다. 다음 [그림 2-3]은 DBSCAN 클러스터링의 예이다[14].



[그림 2-3] DBSCAN 클러스터링

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

반경 내 점의 수(minPts)가 4라고 하면 핵심 벡터 P를 중심으로 반경 내에 인접 벡터인 빨간 점이 최소 점의 개수 이상 있으면 하나의 군집으로 인식한다.

P1의 경우는 최소 점의 개수가 3개이므로 군집의 중심이 되지는 못하지만 P의 군집에 속하기 때문에 이를 경계점이라고 한다.

P4의 경우 어떤 점을 중심으로 하더라도 범위에 포함되지 않게 되는데 이를 노이즈라 한다. 즉, 점을 중심으로 반경 내에 최소 점의 개수 이상의 점이 있으면 그 점을 중심으로 군집을 형성하고 다른 군집과 겹치면 그 군집을 서로 연결한다.

이러한 이유로 DBSCAN의 클러스터링 결과는 원형 모양이 아닌 다양한 형태의 군집이 형성된다[15]. 본 논문에서는 의미적으로 비슷한 단어들인 유의어들의 정확한 판별을 위해서 DBSCAN 알고리즘을 사용하였다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

D. 유사도 측정

유사도 측정은 두 문서 사이의 관련 정도를 수치적으로 계산하는 것으로 벡터로 이루어진 문서를 계산하기 때문에 계산식 역시 벡터 계산으로 이루어지며, 간소화와 정규화 기능을 갖고 있다[16].

코사인 유사도는 두 문서 벡터 간의 각의 코사인 값을 이용하여 유사도 측정을 계산한다. 결과 값은 0에서 1사이의 결과 값이 나오고 1인 경우는 완전히 같은 문서인 경우, 0은 두 문서에서 공통된 단어가 없는 문서를 뜻한다.

다음 식 (3) 은 코사인 유사도 계산식을 나타낸다.

$$\cos(di, dj) = \frac{\sum_{k=1}^{n} w_{ik} \cdot w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2 \cdot \sum_{k=1}^{n} w_{jk}^2}} \qquad (3)$$

이시화의 연구에서는 기존 태그 기반 시스템들은 단일 태그 매칭을 통해 정확도가 낮은 검색 결과를 제공하는 문제점을 해결하기 위해 태그 빈도수 추출 모듈, 태그 유사도 추출 모듈, 태그쌍 가중치 행렬 생성 모듈, 태그 랭킹 모듈을 통해 태그 랭킹 시스템을 구축하여 성능을 향상 시키는 연구를 진행하였다[17].

박수진의 연구에서는 태그 클러스터링 시스템, 소셜 네트워크 시스템, 웹 콘텐츠 랭킹 시스템을 통해 연관된 태그를 추출하고 북마크 수와 다른 사용자의 유사도를 활용한 방법으로 연구를 진행하였다[18].

김혜숙의 연구에서는 문서 간 유사도를 측정하기 위해 single-term 방법으로 색인을 추출하고 색인어를 생성하여 두 문서간의 유사도 측정치, 일치하는 키워드 수, cosine유사도 측정치, threshold값을 넘는 키워드 수의 4가지 방법을 적용하여 유사도를 측정하였다[19].

Collection @ chosun

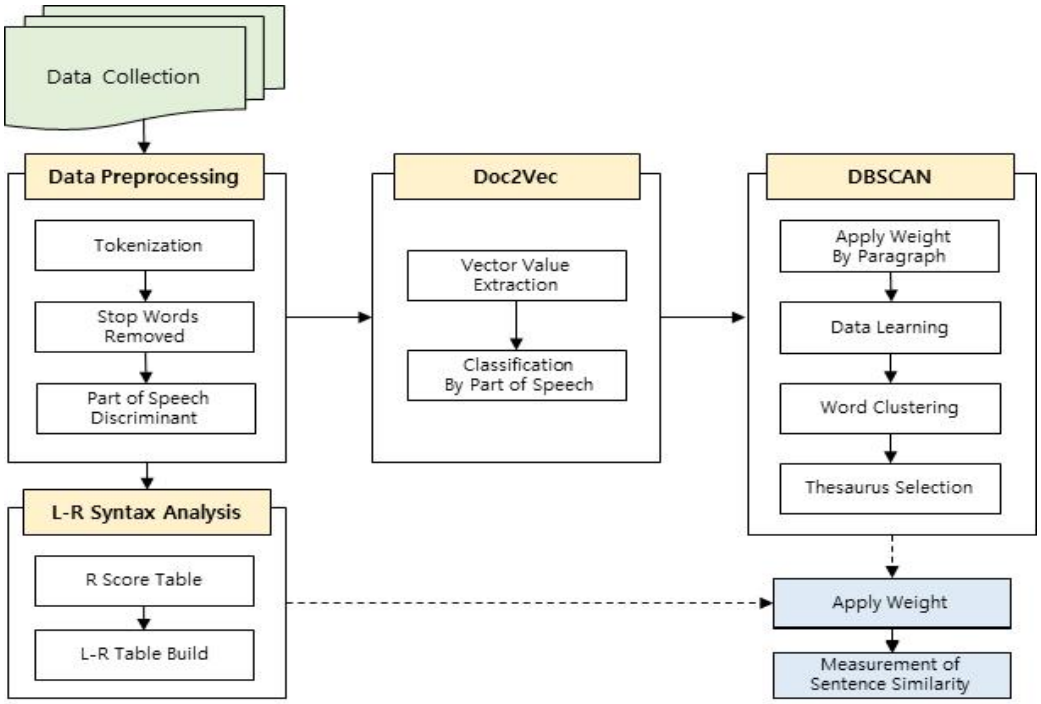
조선대학교
CHOSUN UNIVERSITY

Ⅲ. 유의어 가중치를 적용한 유사도 측정

본 논문에서는 표절 검사 정확성의 향상을 위해 Doc2Vec을 사용하여 벡터값을 구하고 논문의 변형 TF-IDF 알고리즘을 통해 논문의 단락별 가중치를 적용하고 DBSCAN으로 유의어를 찾아 거리 값을 추출한다. L-R 구문분석을 통해 명사 점수를 측정하여 가중치 값을 적용하여 유사도를 측정한다.

A. 시스템 구성도

[그림 3-1]은 본 논문에서 제안하는 시스템 구성도이다. 시스템 구성도는 크게 벡터값을 추출하는 단계와 단어 군집화 단계 L-R 구문분석 단계 및 문서 유사도 측정 단계로 구성된다.



[그림 3-1] 시스템 구성도

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

실험에 사용된 데이터는 학습모델에 사용하기 위한 ‘텍스트 마이닝’을 주제로 한 500편의 논문을 실험 데이터로 ‘TF-IDF’를 주제로 한 100편의 논문을 사용하였다.

데이터는 전처리 과정을 통해 품사를 판별하고 Python 기반 Doc2Vec을 사용하여 벡터값을 추출한다. 벡터값을 구하는 과정에서 단순히 단어의 출현 빈도수로 파악하게 되면 단락에서의 중요 단어가 전체 문서에서 빈도수가 낮아지는 문제점이 있다. 따라서 논문의 단락별 가중치를 적용한 변형 TF-IDF 알고리즘을 적용한다.

DBSCAN 알고리즘을 사용하여 단어별 군집화를 생성하고 유의어를 찾고 L-R 구문분석을 통하여 L-R Table을 구축한다. 군집의 대표 단어와 유의어 사이의 거리 값과 명사 점수를 가중치로 적용하여 유사도를 측정한다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

B. 유의어 추출

본 절에서는 논문 데이터를 전처리 과정 후 Doc2Vec를 이용하여 벡터화를 진행하고 DBSCAN으로 군집화를 진행하여서 유의어를 추출하는 방법에 관해 기술한다.

1. 전처리 과정

PDF 파일 형식으로 수집된 논문 데이터를 텍스트 파일로 변환하고 Python 기반 KoNLPy와 NLTK 라이브러리를 사용하여 문장을 어절 단위로 토큰화를 진행 후 문장의 불용어를 제거한다[20]. 불용어는 모든 영단어를 소문자화 하고 반복적으로 사용된 단어 혹은 특수기호, 인터넷 용어 등을 제거한다.

유의어를 추출하기 전에 단어의 품사를 구분하기 위해서 품사 판별을 진행한다. 품사 판별은 카이스트 태그셋 69개 중 간편한 분석 결과를 위해 22개의 품사 태그를 사용한다. 다음 [표 3-1]은 품사 판별에 사용된 품사 목록을 나타낸다.

[표 3-1] 품사 테그셋

기호	품사	기호	품사
NC	보통 명사	JC	격조사
NQ	교유 명사	JX	보조사
NB	의존 명사	JP	서술 격조사
NP	대명사	EP	선얼말어미
NN	수사	EC	연결어미
PV	동사	ET	전성어미
PA	형용사	EF	종결어미
PX	보조 용언	XP	접두사
MM	관형사	XS	접미사
MA	부사	S	기호
I I	감탄사	F	외국어

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

2. Doc2Vec

단어 군집화를 진행하기 위해서 전처리 과정을 거친 데이터를 Python 기반 Gensim 패키지의 Doc2Vec 모델을 사용하여 벡터값을 추출한다. 학습데이터를 학습 후 각 실험데이터의 각 단어들을 입력 데이터로 입력하여 유사한 벡터값을 추출한다. 다음 [그림 3-1]는 학습 결과를 확인하기 위해 문서 A에서 단어 '유사도'의 벡터값 결과 예이다.

```
101 temp = data.ix[:, 2],map
102   (lambda x: unicode(x).split(','))
103 model = Doc2Vec(temp, size=12)
104
105 print(model[u"유사도"])
```

```
Console 1/A
[0.61551126, -5.52147896, 1.41564536, 0.7643053,
-4.86944265, -2.67892565, 3.11314589, 0.96200034,
-8.23633956, 0.13257686, -0.26589543, 0.24694857]
```

[그림 3-1] 단어 벡터값

추출된 벡터값은 R기반 DBSCAN에 사용하기 위해 [그림 3-2]와 같이 .csv 파일로 저장한다.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	0.89366	0.679926	0.277064	-0.12991	0.395264	0.048666	0.713812	0.149153	0.345863	0.145933	-0.81657	0.693115	0.404186	-0.95167	-0.1794	0.376073	-0.22171	0.115039
2	0.28917	-0.15636	0.181216	-0.25772	0.458959	-0.04138	0.680938	-0.98578	-0.56878	0.948011	0.804267	-0.7992	0.422926	0.653028	0.237125	-0.54232	0.383432	0.748215
3	0.058816	0.634221	0.868422	-0.67844	0.887439	0.136802	-0.52394	0.268389	-0.22878	0.329276	0.056363	-0.70136	0.859112	-0.5972	0.644304	0.810844	-0.95355	0.011293
4	0.868187	0.333204	0.637458	0.803073	0.704831	0.265936	0.019166	0.252698	0.543018	-0.09407	0.95974	0.737987	0.805013	0.693627	-0.2866	0.121967	-0.43351	-0.82704
5	0.958193	-0.60884	0.045353	0.768628	0.092291	0.907268	0.775305	0.97473	-0.78199	0.281098	0.484258	-0.31259	0.699502	0.492893	0.627772	0.37121	0.505796	-0.27525
6	-0.04567	0.272264	0.487312	0.52065	0.835023	0.657795	0.869651	-0.8153	0.07438	0.441779	-0.83941	0.89918	0.556188	0.474832	0.620246	-0.98927	0.752879	-0.66247
7	0.17913	-0.82685	0.812589	0.35688	0.970303	-0.74675	0.919654	0.855599	-0.58251	0.501083	0.629548	-0.92901	0.542179	0.284758	-0.42892	0.184735	0.298033	0.785455
8	0.416262	-0.96692	-0.0548	-0.60676	0.998718	0.694506	0.399937	0.818323	0.107721	0.031395	-0.30029	0.557617	-0.11636	-0.6511	0.598584	-0.55553	-0.59548	0.414178
9	0.63429	0.318298	0.336538	0.194031	0.191285	0.321588	0.28045	0.297042	0.768571	0.339555	0.676083	-0.67018	0.839929	0.871199	0.384714	0.963116	0.027517	-0.99403
10	0.052078	-0.30209	-0.72533	0.815363	0.692695	0.908644	-0.96107	0.867724	0.490534	0.394248	0.288401	-0.99222	0.981181	0.375996	0.827477	0.522954	0.74833	0.286292
11	0.492615	0.449872	0.826566	0.459682	0.010663	0.234977	0.105563	0.497249	0.877855	-0.935	0.889269	0.118509	0.19007	0.979527	0.25971	0.247744	0.153654	-0.16889
12	0.762229	0.222117	0.750635	0.305025	0.692549	-0.63502	0.963012	0.366458	0.896618	0.70978	0.702323	0.908369	0.5537	0.6323	-0.1266	-0.98348	0.163776	0.428504
13	0.989177	0.541529	-0.9477	-0.81167	0.25148	0.385167	0.581251	0.362842	-0.29428	-0.08524	-0.84149	-0.80009	-0.54898	0.209415	0.107153	0.87945	0.204661	-0.12525
14	0.641243	0.129721	0.551498	0.980449	0.444898	0.615474	-0.61809	0.494411	0.260907	0.54812	-0.69888	0.87479	0.969161	0.527957	0.567601	-0.96522	-0.45826	-0.84981
15	0.426627	0.784515	0.761509	0.5323	-0.51988	0.335451	0.57235	-0.92284	0.141884	0.858208	0.633661	0.60468	0.809601	-0.29194	-0.48358	0.933838	0.961344	0.665539
16	0.601206	-0.59034	-0.17554	0.834893	0.00283	0.376617	0.944901	-0.18039	-0.07769	0.750386	0.907081	0.557888	-0.29205	0.973973	0.598991	0.833363	-0.4104	0.065097
17	0.351289	0.953553	0.744865	0.687506	-0.14032	-0.5544	0.277689	0.666284	0.459504	-0.54613	0.20001	0.38309	-0.50716	0.24451	-0.15263	0.675186	-0.43233	-0.26174
18	0.715259	0.014229	0.619705	-0.22716	0.369021	0.840472	0.306713	0.2442	0.662994	0.024346	0.902783	-0.65908	0.768115	0.246121	0.788083	0.068986	0.398609	0.722987
19	-0.51115	-0.89861	0.921452	-0.26208	-0.53966	-0.26837	0.792757	0.501964	0.297709	0.166418	-0.93932	0.178414	0.508866	-0.56571	0.878789	0.276458	0.907352	0.561215
20	0.929394	0.571614	0.852155	0.823018	-0.80079	0.517186	-0.35157	0.428813	-0.59971	-0.87757	0.217209	0.967062	-0.9051	0.47304	0.867065	0.701592	-0.84454	0.709084
21	0.675666	-0.2927	0.174096	0.939936	0.378126	0.63584	0.648005	0.247276	-0.5563	-0.17176	0.365586	0.844864	0.157075	0.480968	-0.86974	0.828311	-0.71899	-0.8106
22	0.14786	-0.8571	0.923932	-0.22307	0.906221	0.518924	-0.28166	0.367836	0.633303	0.720434	-0.06376	0.653692	0.784876	0.8189	-0.35734	0.125479	0.587721	0.60432
23	-0.60282	0.031419	0.583807	0.172227	-0.65109	0.213571	0.495746	-0.25788	0.279365	0.435143	0.98521	-0.3436	0.994926	-0.29655	-0.27142	0.295858	-0.12031	0.323804
24	0.338929	0.939121	0.225425	0.80774	0.653395	-0.94361	0.629358	0.92341	0.928622	0.426436	-0.5702	0.081136	0.497119	0.742645	0.756485	-0.93852	0.014488	0.628607
25	0.597118	-0.24296	0.5726	0.034378	0.066289	-0.08466	0.784262	0.568302	0.344499	0.710766	0.002743	0.952647	0.726693	-0.08682	0.359428	0.229189	0.049824	-0.19006
26	0.807863	0.173601	0.840453	0.460666	0.515227	0.190385	-0.5781	0.295174	0.835347	-0.60681	0.692346	-0.65333	-0.69822	0.513694	0.131023	-0.16565	0.819602	0.607655
27	0.62103	-0.50999	0.68758	0.584949	0.918507	0.869199	0.591754	-0.37991	-0.5818	0.510052	0.986054	0.148803	0.889666	0.641681	0.844262	0.114161	0.480878	0.332289
28	0.392914	0.595026	0.814106	-0.31315	0.067891	0.512177	0.906784	0.027074	0.385957	0.715975	0.169114	0.51042	-0.19432	0.045988	0.783291	0.275906	-0.47146	0.899707
29	0.832197	0.892028	0.928233	-0.88961	0.458198	0.791701	0.692861	-0.19202	0.602791	0.058075	0.841272	0.008263	0.632258	0.023344	0.745273	0.427902	-0.25477	0.88334
30	0.821854	0.06704	0.601545	0.235035	-0.22101	0.683222	0.443053	0.963817	0.571065	-0.16523	0.442781	0.403913	-0.46262	0.848379	0.755721	-0.69065	0.106216	0.581594
31	0.805347	0.704628	0.684799	0.053072	0.52536	0.861688	0.400751	0.90802	0.425608	0.744998	0.611	0.053325	0.996765	0.48869	0.994126	0.121737	-0.73232	0.809541
32	0.821656	-0.98352	0.874123	0.395969	0.434185	0.504089	0.447665	0.588094	-0.34183	-0.91592	0.690825	0.088719	0.901458	0.445649	0.435124	-0.5357	0.704116	0.023006
33	0.402972	0.813132	0.41433	0.670586	0.735401	-0.32771	0.779467	0.249458	0.253894	0.031967	-0.34873	0.388782	0.055735	0.238248	0.339397	0.290515	0.326338	0.704604

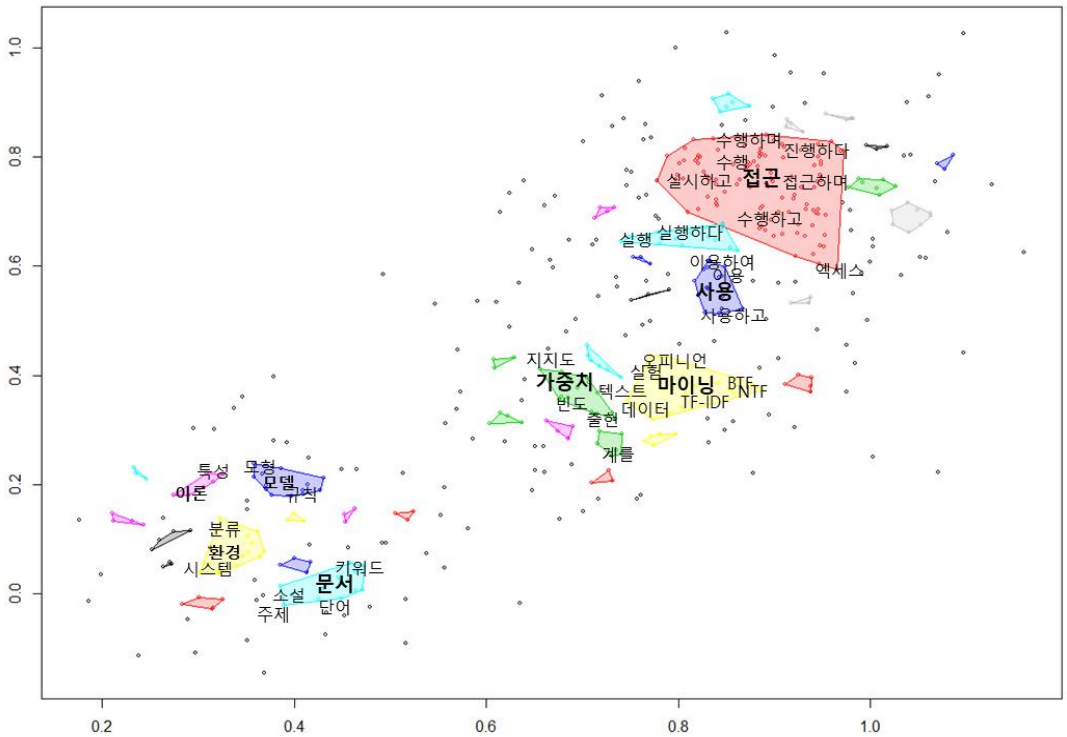
[그림 3-2] 벡터값 .csv 파일 저장

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

3. DBSCAN 군집화

추출된 벡터값을 단락별 가중치를 적용해서 R기반 DBSCAN으로 단어 군집화를 진행한다. 단락별 가중치가 군집화 결과에 영향이 있는지 확인하기 위하여 가중치를 적용하지 않은 경우와 적용한 경우의 군집화를 비교한다. 각 군집의 중심점에 위치한 단어를 대표 단어로 선정하고 군집내의 단어들을 유의어로 지정한다. [그림 3-3]는 가중치를 적용하지 않은 군집이고 [그림 3-4]은 가중치를 적용한 군집의 예이다.



[그림 3-3] DBSCAN 군집화 (단락별 가중치 적용하지 않음)

대표단어의 예로는 '접근, 사용, 마이닝, 가중치, 문서, 환경' 등이 선정되고 각 대표단어 주변의 유의어로 군집이 형성된다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

4. 단락별 가중치

선행 연구 결과를 통하여 기존의 6어절 이상이 동일한 경우에 유사도를 검사할 경우에 다른 의미의 단어이지만 유사도 검사에 검출 되는 경우를 보안하기 위하여 단락별 가중치를 부여한다[21].

예를 들어 한 문서에서 서론, 관련 연구에서는 A라는 알고리즘 단어가 많이 등장하나 본론에서는 A알고리즘을 변형한 A1이라는 알고리즘이 등장하는 경우가 있다. A1과 A의 어절 순서, 혹은 비슷한 단어로 이루어졌을 경우 유사도 검사에 검출 되는 경우가 있으며, 해당 문서에서 가장 중요한 단어는 A가 아니라 A1임을 알 수 있다.

따라서 본 논문에서는 서론, 관련 연구, 본론, 결론의 큰 단락 별 단어의 TF-IDF에 단락별 가중치를 추가하여 적용한다. 식 (3) 은 가중치 적용 식을 나타낸다[22].

$$W_{(t,j,d)} = tf_{(t,j,d)} \times idf_{(t)} \times s_j \tag{3}$$

- $W_{(t,j,d)}$: 문서 d 내의 단락 j 에서의 단어 t의 가중치
- $tf_{(t,j,d)}$: 문서 d 내의 단락 j 에서의 단어 t의 빈도수
- s_j : 단락 j 에 대한 가중치

단락에 대한 가중치 s_j는 다음 식 (4), 식 (5) 와 같이 계산하였다.

$$S_j = \frac{f_{(tn,j)} \cdot w_l}{f_{(t_n,d)}} \tag{4}$$

- $f_{(t_n,d)}$: 문서 d 내의 명사 단어 t_n의 개수
- $f_{(t_n,j)}$: 단락 j 내의 명사 단어 t_n의 개수
- w_l : 문장 가중치

Collection @ chosun

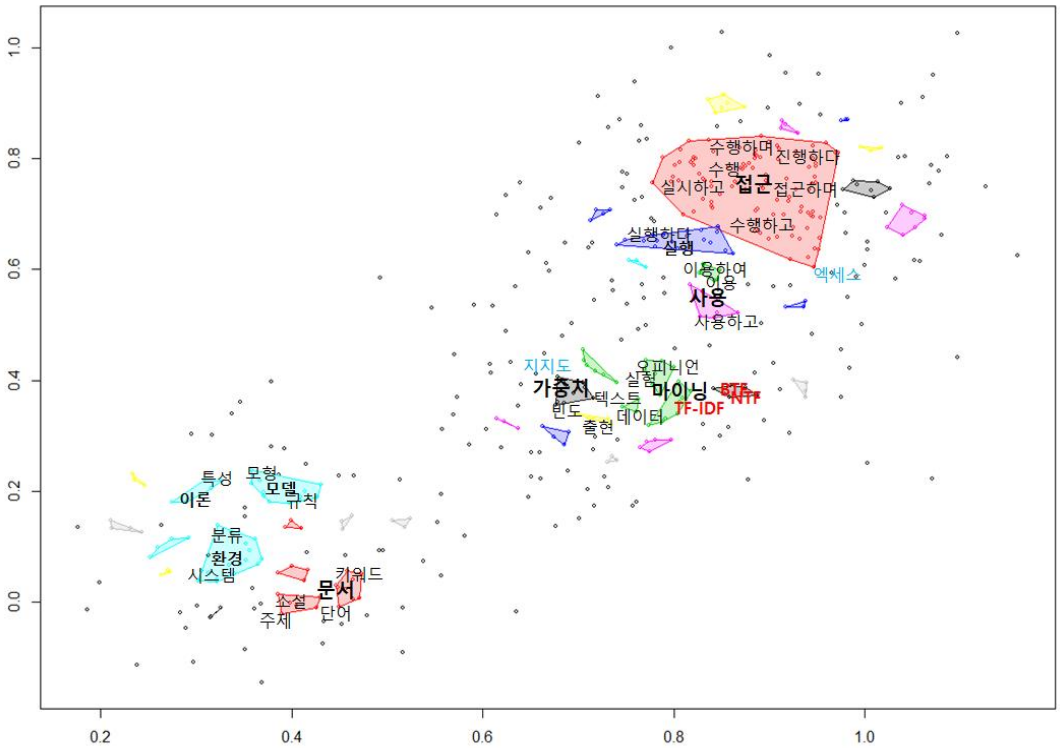
조선대학교
CHOSUN UNIVERSITY

$$w_l = \frac{lf_{(t_l,j)}}{lf_{(t_l,p)}} \tag{5}$$

- $lf_{(t_l,j)}$: 단락 j 내의 문장 t_l의 개수
- $lf_{(t_n,p)}$: 전체 문서 p 에서의 문장 t_l의 개수

'텍스트 마이닝' 관련 논문 500편을 기준으로 계산하였을 때 서론은 0.2 , 관련 연구 0.15, 본론 0.4, 결론 0.25 의 가중치를 적용한다.

단락별 가중치를 적용하여 각 문서의 단락별 가중치를 적용하여 군집 결과를 추출한다.



[그림 3-4] DBSCAN 군집화 (단락별 가중치 적용)

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

단어별 가중치를 적용하였을 경우 군집이 세분화가 된다. [그림 3-2]의 대표단어 마이닝의 군집이 단락별 가중치를 적용하면 [그림 3-3]처럼 'TF-IDF', 'BTF', 'NTF' 등의 단어가 하나의 군집으로 변형된다. 또한 '엑세스', '지지도'와 같은 단어는 군집에서 제외 되는 것을 확인 할 수 있다. 이는 단락별 가중치가 군집형성에 영향을 미치는 것을 알 수 있다.

위의 결과를 통해 거리 가중치를 측정한다. 단순히 대표 단어와 유의어 간의 거리 값을 가중치로 사용하기에는 군집 형태가 일정하지 않으므로 식 (6) 와 같이 거리 가중치를 측정한다.

$$W_{(j,i)} = \frac{rw_j - sw_i}{t} \tag{6}$$

- $W_{(j,i)}$: 거리 가중치
- rw_j : 대표단어 위치 값
- sw_i : 유의어 위치 값
- t : 군집 최대 거리 값

다음 [표 3-3]는 문서 A의 단락별 가중치를 적용한 군집 단어의 거리 값의 예이다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[표 3-3] 문서 A의 각 단락별 거리 값 예

서 론		관련 연구		본 론		결 론	
단어	거리값	단어	거리값	단어	거리값	단어	거리값
TF-IDF	0.03593	알고리즘	0.02906	ETF	0.01821	결과	0.02625
문서	0.06844	TF-IDF	0.03022	TF-IDF	0.01729	ETF	0.04365
뉴스	0.07255	키워드	0.04326	JTF	0.02239	단어	0.06792
가중치	0.09216	단어	0.0645	TF	0.03012	TF-IDF	0.01092
키워드	0.01355	가중치	0.07222	IDF	0.07505	리스트	0.01302

문서 A의 단어 중 가장 높은 순위의 단어는 TF-IDF 이다. 단락별 가중치를 적용하였을 경우 [표 3-3]과 같이 서론에서는 TF-IDF가 가장 높은 순위의 단어이나 본론에서는 ETF라는 변형된 알고리즘의 단어가 가장 높은 순위의 단어이다. 이처럼 단락별로 표절검사의 중요도가 높은 곳을 알 수 있다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

C. L-R 구문분석

DBSCAN 군집의 대표 단어와 유의어의 거리 값은 값의 범위가 적으므로 유사도 판별 가중치로 사용에 어려움이 있다. 본 논문에서는 유사도 판별의 정확성을 높이기 위해서 L-R 구문분석을 통하여 L-R Table을 구축하고 명사 점수를 유사도 판별 가중치로 적용한다.

한국어의 구조 특성상 'L + R' 구조로 '명사 + 조사'의 형태로 이루어진다. 조사로 이루어진 R의 명사 가능 점수를 통해 L의 명사 점수를 측정하여 가중치로 적용한다. R의 명사 가능 점수는 세종말뭉치를 이용하여 학습된 명사 가능 점수 데이터를 사용하였으며, [표 3-4] 학습된 R 데이터의 예와 같이 구성되어 있다[23].

[표 3-4] R 데이터 예

단 어	명사 가능 점수
했다	1.000000
하게	-0.926182
하며	0.327843
하는	0.079298
였다면	-1.000000

명사 가능 점수의 범위는 최소 -1 , 최대 1의 범위의 점수를 가지게 된다. 이러한 명사 가능 점수를 이용하여 명사 점수를 계산하는 방법은 다음 식 (7) 와 같다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

$$S_L = \frac{(R_{N1} \times S_r) + \cdots + (R_{Nn} \times S_r)}{R_N} \tag{7}$$

S_L은 명사 점수이고 R_{Nn}은 n번째 R의 등장 횟수 , S_r은 R의 점수 , R_N은 R의 총 등장 횟수를 의미한다. 만약 '접근+했다' 가 5번 , '접근+하게' 가 2번 등장하였다면 적용의 명사 점수는 식 (8) 와 같이 0.4497의 점수를 가지게 된다.

$$\frac{(5 \times 1.0 + 2 \times (-0.926))}{7} = 0.4497 \tag{8}$$

· S_L : 명사 가능 점수 · R_{Nn} : n번째 R의 등장 횟수
· S_r : R의 점수 · R_N : R의 총 등장 횟수

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

Ⅳ. 실험 및 결과

본 장에서는 Ⅲ장에서 제안한 DBSCAN을 활용한 문서 유사도 검사 방법에 대한 평가를 위해 텍스트 마이닝 관련 논문 500편 과 TF-IDF 관련 논문 100편으로 실험을 진행 하였으며, 제안한 방법의 유사도 결과와 성능을 평가한다.

본 연구에서 실험에 사용한 환경은 [표 4-1]과 같다.

[표 4-1] 실험 환경

Division		Contents
HW	CPU	Intel(R) Core(TM) i5-720U CPU @ 2.00GHZ
	GPU	Intel(R) HD Graphics 550
	RAM	4GB
	HDD	500GB
SW	OS	Windows10
	Program	Python, R studio

A. 유사도 측정

문서의 유사도를 평가하기 위해서 문장 유사도의 기준을 구한 뒤 기존의 유사도 측정 방법과 유사도 평가를 진행한다.

문장 유사도의 기준은 학습모델에 사용된 '텍스트 마이닝' 관련 논문 500편을 이용하여 문장 유사도의 기준값을 구한다. 유사도 기준은 0.5부터 0.9의 기준들에 따른 정확률과 정확율, 조화평균을 통해 정확도를 확인하여 문장 유사도의 기준값을 정한다.

다음 식 (9) 과 식 (10) 은 정확률과 재현율의 계산식이다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

$$정확률(P) = \frac{tp}{tp + fp} \tag{9}$$

$$재현율(R) = \frac{tp}{tp + fn} \tag{10}$$

· tp : 시스템이 유사한 문장이라고 제시한 문장 중 맞는 문장 수
· fp : 유사한 문장이라고 제시한 문장 중에 틀린 문장 수
· fn : 유사한 문장이 아니라고 제시한 문장 중 유사한 문장 수

또한 정확률과 재현율은 반비례 관계이기 때문에 조화평균(F-measure)을 계산하여 문장유사도의 기준값에 대한 정확도를 평가한다. 조화평균 계산은 식 (11) 과 같다.

$$F-measure(F) = 2 \cdot \frac{P \cdot R}{P + R} \tag{11}$$

조화평균 F는 0과 1사이의 값을 가지며, 정확률 P와 재현율 R의 값이 높아야 F 값이 높아진다. 각 기준값은 다음 [표 4-2]와 같은 결과가 나왔다.

[표 4-2] 기준값 측정

기준값	0.5	0.6	0.7	0.8	0.9
정확률	52.9%	74.1%	79.7%	70.5%	62.2%
재현율	33.3%	69.8%	67.3%	52.9%	47.1%
조화평균	48.7%	71.8%	72.9%	60.4%	53.6%

각 기준값에 대한 정확률은 0.7의 경우가 79.7%로 가장 높고 재현율은 0.6의 경우 69.8%로 가장 높으나 조화평균이 0.7의 경우 72.9%로 정확률과 조화평균이 0.7

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

의 경우 가장 높은 결과가 나왔다. 따라서 본 논문에서 유사도의 기준값은 0.7로 정하였다.

유의어 가중치를 적용하는 방법의 성능확인을 위해 문서 A - B의 문장 유사도를 코사인 유사도로 계산한다[24]. 문서 A는 군집의 대표단어가 내용에 포함되어 있고 문서 B는 대표 단어는 포함되지 않았지만 유의어가 포함된 문서이다.

유의어에 대한 거리 가중치와 단락별 가중치가 각각 유사도에 어떤 영향을 미치는지 확인하기 위하여 B문서에 거리 가중치만 적용한 B1과 단락별 가중치와 거리 가중치를 적용한 B2의 문장 유사도를 판별한다.

유의어에 대한 가중치는 유의어의 거리 가중치와 L-R Table에서의 S_L 점수를 합하여준다. 즉, '문장의 유사도 + 거리 가중치 + 명사 점수'를 합하여 준다.

[표 4-3]은 단락별 가중치를 미적용한 문서 A와 B , 문서 A와 B1의 문장별 유사도 값과 단락별 가중치를 적용한 문서 A와 B2의 유사도 값을 나타낸다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

[표 4-3] 문장별 유사도 비교

	문서 A - B		
	코사인 유사도	단락별 가중치 미적용	단락별 가중치 적용
	A - B	A - B1	A - B2
1	0.628	0.733	0.649
2	0.986	0.391	0.316
3	0.967	0.774	0.333
4	0.326	0.166	0.174
⋮	⋮	⋮	⋮
354	0.713	0.464	1.220
355	0.639	1.430	0.175
356	0.589	0.464	0.796
문서 유사도	39 %	53 %	46 %

문서 A와 B의 문장 수는 각각 356 문장이다. A - B1 과 A - B2에 가중치를 적용하여 문장 유사도 값이 단순히 높아지지 않고 2번 문장의 유사도처럼 높은 문장 유사도 값에서 명사 점수 변동으로 인하여 낮은 유사도 값이 나오고 356번 A - B2 비교문장 유사도처럼 높은 유사도 값으로 결과 값이 나오는 것을 확인 할 수 있다. 이는 DBSCAN을 통해 측정한 거리 값만으로 유사도 측정을 하는 것 보다 구문분석을 통해 유의어가 알맞은지를 비교 후 명사 점수를 적용하여 유사도를 측정하는 것이 좋은 정확도가 나옴을 알 수 있다.

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

B. 유사도 성능 평가

본 논문에서 제안한 방법의 성능을 확인하기 위해 LSA와 N-gram 두 가지 방법을 이용하여 유사도를 평가하는 방법과 성능을 비교 하였다. LSA를 통해 문장별 행렬을 벡터 간의 유사도를 측정하고 음절 bi-gram과 ti-gram을 추출하여 N-gram 중복검사를 통해 LSA 와 N-gram의 임계값이 각각 0.7, 0.5일 때 유사도를 계산한다[25]. 다음 [표 4-4]는 문서 A와 B의 유사도 평가 결과이다.

[표 4-4] 유사도 평가

	코사인 유사도	LSA N-gram	제안 방법
문서 유사도	23.7 %	32.1 %	30.5 %
정확률	63.9 %	88.2 %	94.5 %
재현율	59.1 %	96.0 %	92.7 %
조화평균	0.6140	0.9193	0.9359

문서 유사도는 LSA와 N-gram을 이용하였을 때 32.1% 로 높게 나왔고 제안한 방법의 유사도는 30.5% 가 나왔다. 하지만 정확률과 조화평균은 제안 방법이 더 높은 결과가 나왔다. 이는 [표 4-5] 처럼 유사 문장 내 단어가 비슷한 단어이지만 다른 의미의 단어를 포함하고 있는 문장의 경우 본 논문에서 제안한 방법은 유사도가 낮게 나오기 때문이다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[표 4-5] 유사도 검출 문장 예

	LSA N-gram	제안 방법
원본 문장 예	⋮ 문서 내 단어의 빈도수를 TF-IDF 알고리즘을 사용하여 측정한다. ⋮	
유사 문장 예	⋮ 문서 내 키워드의 빈도수를 TF-IDF 변형알고리즘을 사용하여 측정한다. ⋮	⋮ 문서 내 키워드의 빈도수를 TF-IDF 변형알고리즘을 사용하여 측정한다. ⋮

LSA 와 N-gram을 사용한 유사도 결과는 “TF-IDF 변형알고리즘”이 원본 문장의 “TF-IDF 알고리즘”과 0.7이상의 유사 문장이라는 결과가 나왔으나 본 논문에서 제안한 방밥은 “변형알고리즘”이 유사 단어에서 제외되어 유사 문장에서 0.7 이하의 유사도의 결과가 나오므로 LSA와 N-gram을 이용하여 유사도를 측정한 경우보다 본 논문에서 제안한 방법이 유사도 측정에 정확한 결과를 도출됨을 알 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

V. 결론 및 향후연구

본 논문에서는 논문 표절 검사 정확성의 향상을 위해서 문서들의 유사도를 유의어 가중치를 적용하여 유사도를 측정하는 방법을 제안 하였다. 텍스트 마이닝 관련 논문 500편과 TF-IDF 관련 논문 100편을 실험 데이터로 사용하였다.

PDF 파일 형식의 논문들을 수집하여 텍스트 파일로 변형하고 Python 기반 KoNLPy와 NLTK 라이브러리를 사용하여 전처리 과정을 진행하였다. 전처리 과정을 거친 데이터는 Doc2vec으로 단어의 벡터값을 추출하여 csv파일로 저장하여 DBSCAN 군집화에 사용한다.

DBSCAN으로 군집화를 진행하고 군집내의 중심 단어를 대표단어로 설정하고 해당 군집 안에 다른 단어들을 유의어로 선정하여 대표 단어와 유의어 간 거리 값을 가중치로 적용하였다.

선행 연구에서 기존의 6어절이 동일한 경우에 유사도 검사를 할 경우 유사한 의미이지만 다른 의미의 단어가 포함된 문장이 유사도 검사에 검출 되어 이를 보안하기 위해 서론, 관련 연구, 본론, 결론의 단락별 가중치를 적용하고 군집화를 진행하였다. 단락별 가중치는 기존의 TF-IDF 가중치 계산식을 변형하였으며 군집의 결과가 달라짐을 알 수 있었다. 하지만 거리 값만으로 가중치를 적용하기에는 수치가 낮아 L-R 구문분석으로 L-R Table을 구축하고 R의 명사 가능 점수로 L의 명사 점수를 구하여 문장 유사도 판별의 유의어 가중치에 적용하였다.

유사도를 측정하기 전에 유사도의 기준을 선정하기 위해서 0.5부터 0.9 까지 기준에 대해 정확률, 재현율, 조화평균을 활용하여 유사 문장 판별의 정확도로 0.7의 유사도 기준을 선정하였다. 문장별 유사도는 '문장 유사도 + 거리 가중치 + 명사 점수'로 계산하여 가중치를 적용하지 않고 코사인 유사도로 문장 유사도를 측정한 문서 A - B의 경우와 B문서에 단락별 가중치를 미적용한 유의어 가중치를 적용한 문서 A - B1, 단락별 가중치를 적용한 문서 A - B2의 유사도를 측정 후 비교한 결과 단락별 가중치를 적용한 경우 다른 의미의 단어를 사용한 경우 유사도가 낮게 나옴을 알 수 있었다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

유사도 성능 평가를 위해 LSA와 N-gram을 이용한 방법과 제안 방법의 유사도 결과를 비교 하였다. 비교 결과 LSA와 N-gram을 사용한 경우 문서 유사도는 32.1% 제안한 방법은 30.5%의 결과가 나왔고 조화평균은 제안한 방법이 0.9359로 가장 높은 결과가 나왔다. 이는 비슷한 단어지만 다른 의미인 단어의 유사도를 측정하는 경우 본 논문에서 제안한 방법이 정확한 성능이 나타나는 것을 알 수 있었다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

참고문헌

[1] I. Hwang. “Development of A Plagiarism Detection System Using Web Search and Morpheme Analysis”, Journal of Information Technology Applications & Management, Vol. 16, No. 1, pp. 21-36, 2009.

[2] Lee In Jae. “Series 3. Plagiarism and Proper Citation”, Korean journal of aesthetics and cosmetics society, Vol. 10, No. 4, pp. 739-745, 2012.

[3] D. Kwack. “A Study on the Types of Plagiarism and Appropriate Citation Practices of Writing Research Papers″, Journal of the Korean Society for Library and Information Science, Vol. 41, No. 3, pp. 103-126, 2007.

[4] Naver Knowledge Encyclopedia, 2018.5.7, http://terms.naver.com/entry.nhn?docId=1315211&cid=40942&categoryId=31693

[5] R. Robertson. “Understanding inverse document frequency: on theoretical arguments for IDF”, Journal of Documentation, Vol. 60, No. 5, pp. 503-520, 2004.

[6] Y. Lee, S. Lee. “A Research on Enhancement of Text Categorization Performance by using Okapi BM25 Word Weight Method”, Journal of the Korea Academia Industrial cooperation Society, Vol. 11, No. 12, pp. 5089-5096, 2010.

[7] Z. S. Harris. “Distributional structure”, Word, vol. 10, pp.146-162, 1954.

[8] P. D. Turney, P. Pantel. “From frequency to meaning: Vector space models of semantics”, Journal of Artificial Intelligence Research, vol. 37, pp. 141-188, 2010.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean. “Distributed representations of words and phrases and their compositionality”, in Proceedings of the Conference and Workshop on Advances in Neural Information Processing Systems, 2013, pp. 3111-3119.

[10] Q. Le, T. Milokov. “Distributed Representations of Sentences and Documents”, Proceedings of the 31st International Conference on Machine

Collection @ chosun

조선대학교 CHOSUN UNIVERSITY

Learning, 2014.

[11] K. Cheng, J. Li, J. Tang, H. Liu. “Unsupervised Sentiment Analysis with Signed Social Networks”, Proc of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 777-786, 2017.

[12] D. Kim, M. Koo. “Categorization of Korean News Articles Based on Convolutional Neural Network Using Doc2Vec and Word2Vec”, Journal of KIISE, Vol. 44, No. 7, pp. 742-747, 2017.

[13] J. Han, M. Kamber, J. Pei. “Data Mining: Concepts and Techniques”, Morgan Kaufmeann, 2005.

[14] M. Kwon, Y. Kang, H. Han, D. Cho. “Adaptive DBSCAN for time-varing clustering DBSCAN”, Information and Control Symposium, Vol. 2016, No. 4, pp. 134-135, 2016.

[15] T. Kim, J. Chung. “A Study on the Identification of Accident Hot Spots Using DBSCAN : Focused on Gyeong-Bu Expressway”, Journal of Transport Research, Vol. 21, No. 3, pp. 55-63, 2014.

[16] Cho Sung Woong. “Implementation of a Creation Control System for Collaboration Documents Based on Keyword Similarity Analysis”, Degree paper (Master's) -- Graduate School of Graduate School of South Jeolla University : Department of Computer Science 2003. 2

[17] S. Lee. Hwang. “Tag Ranking System based on Semantic Similarity of Tag-pair”, Journal of Korea Multimedia Society, Vol. 16, No. 11, pp. 1305-1314, 2013.

[18] S. Park, S. Lee, D. Hwang. “A Web Contents Ranking System using Related Tag & Similar User Weigh”, Journal of Korea Multimedia Society, Vol. 14, No. 4, pp. 567-576, 2011.

[19] H. Kim, S. Park, S. Kim. “A Research of Documents Similarity Measuring Based on Word Weight”, Journal of Korea Multimedia Society, pp. 198-201, 2003.

[20] E. Park, S. Cho. “KoNLPy: An Easy and Concise Korean Information Processing Python Package”, Human and Language Technology, Vol. 26, pp. 133-136, 2014.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[21] Byeong-sik Kim, Ju-hyun Shin. “Similarity discrimination method based on thesaurus translation for avoiding document plagiarism”, Korea Multimedia Society Fall Conference Proceedings, Vol. 20, No. 2, 2017

[22] Heung-Nam Kim, Kee-Sung Lee, Geun-Sik Jo. “Document Classification using Weighted Associative Classifier”, Journal of the Korea Institute of Information Security & Cryptology, Vol. 30, No. 2, pp.154-156, 2003.10

[23] Yoo-Hun Won. “Efficient LR(k) Parsing Algorithms”, Master’s Thesis of Korea Advanced Institute of Science, 1975.

[24] M. Kim, S. Lee. “Measures of Abnormal User Activities in Online Comments Based on Cosine Similarity”, Journal of the Korea Institute of Information Security & Cryptology, Vol. 24, No. 2, pp. 335-343, 2014.3

[25] Hye-Sung Ji, Joon-Hee Joh, Heui-seok Lim. “A Detection Method of Similar Sentences Considering Plagiarism Patterns of Korean Sentence”, Journal of Korea Computer Education Association, Vol. 13, No. 6, pp.78-89, 2010.11

Collection @ chosun