Journal of Korea Multimedia Society Vol. 21, No. 9, September 2018(pp. 1035-1043)
https://doi.org/10.9717/kmms.2018.21.9.1035

DBSCAN을 활용한 유의어 변환 문서 유사도 측정 방법

김병식†, 신주현††

A Method for Measuring Similarity Measure of Thesaurus Transformation Documents using DBSCAN

Byeongsik Kim†, Juhyun Shin††

ABSTRACT

There is a case where the core content of another person's work is decorated as though it is his own thoughts by changing own thoughts without showing the source. Plagiarism test of copykiller free service used in plagiarism check is performed by comparing plagiarism more than 6th word. However, it is not enough to judge it as a plagiarism with a six - word match if it is replaced with a similar word. Therefore, in this paper, we construct word clusters by using DBSCAN algorithm, find synonyms, convert the words in the clusters into representative synonyms, and construct L-R tables through L-R parsing. We then propose a method for determining the similarity of documents by applying weights to the thesaurus and weights for each paragraph of the thesis.

Key words: Doc2Vec, DBSCAN, Weight by Paragraph, Thesaurus Weight, Measure Similarity

1. 서 론

인터넷의 발달로 다양한 정보를 시간과 장소에 제약되지 않고 쉽게 구할 수 있다. 하지만 이러한 정보를 악용하는 단점이 발생하게 되는데 그중 하나가 표절이다. 소설, 모바일게임, 드라마 등 여러 분야에서의 표절의혹이 있으며 논문도 표절 사례가 존재한다[1]. 예로는 타인의 논문 핵심 내용을 출처를 밝히지 않고 사용하는 경우가 있다. 타인의 생각이나 문장, 단락, 그림, 표 등을 사용하게 되면 반드시 그 출처를 명확하게 밝혀야 하는데 자신의 생각으로 말을 바꿔 쓰면서 꾸미는 경우가 존재한다[2]. Copykiller 무료 서비스의 표준 처리 기준을 보면 6어절 이상 동일한 경우를 표절로 판단한다. 하지만 말을 바꿔 쓰는 것에 있어 6어절 이상이 동일한 경우를 표절로 판단하기에는 부족하다고 판단된다. 따라서 본 논문에서는 TF-IDF 관련 논문을 수집하여 Python 기반의 KoNLPy와 NLTK를 사용하여 불용어를 제거한 후 HannanumPosTagger를 통해 품사를 판별하여 데이터로 사용하였다. 논문의 서론, 관련 연구, 본론, 결론 마다 가중치를 부여한 TF-IDF 알고리즘을 적용하여 벡터값을 찾아 DBSCAN알고리즘으로 단어의 군집을 생성하여 유의어를 찾는다. L-R 구문분석을 통해 L-R Table을 구축하고 유의어에 대한 가중치를 적용해서 문장의 유사도를 측정하는 방법을 제안한다.

※ Corresponding Author : Juhyun shin, Address: (61452) Pilmun-daero 209, Dong-gu, Gwangju, Korea, TEL : +82-62-230-7162, FAX : +82-62-233-6896, E-mail : jhshinkr@chosun.ac.kr
Receipt date : Apr. 9, 2018, Revision date : Jun. 11, 2018
Approval date : Jun. 20, 2018

† Dept. of Software Convergence Engineering Chosun University
(E-mail : eunbesu@gmail.com)
†† Dept. of ICT Convergence, Chosun University
※ This study was supported by research fund from Chosun University, 2018

본 논문의 구성은 다음과 같다. 2장에서는 Doc2Vec와 DBSCAN에 대하여 설명하고, 3장에서는 본 논문에서 제안하는 방법을 설명한다. 4장에서는 제안한 방법을 적용해서 문장의 유사도를 판별한다. 5장에서는 실험 결과와 향후 연구 방향을 기술한다.

2. 관련연구

2.1 TF-IDF

TF-IDF는 문서 간 단어 가중치를 수식 1과 같이 단어 빈도(Term Frequency)와 역 문서 빈도(Inverse Document Frequency)의 곱으로 나타내고 단어를 포함한 문서의 빈도를 고려해 가중치를 측정하는 방법이다[3].

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

$tf_{(t,d)}$는 문서 d 내의 단어 t의 가중치이다. $idf_{(t,d)}$는 단어 t가 출현하는 문서 빈도에 역을 취하므로 t의 출현빈도가 높을수록 많이 쓰이는 단어이므로 가중치 값이 적게 적용된다. 역 문서 빈도는 다음 수식 2와 같다.

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

N은 전체 문서 개수이며, $idf_{(t)}$는 로그를 취하여 값이 적게 나오므로 TF-IDF 값은 $tf_{(t,d)}$의 영향을 많이 받는다[4].

2.2 Doc2Vec

Doc2Vec에는 Fig. 1과 같이 distributed memory model(DM) 과 distributed bag of words(DBOW) 두 모델이 있으며 가변 길이의 텍스트를 고정 길이의 벡터로 표현하는 비지도 학습 알고리즘이다[5].

DM 모델은 벡터와 주변 단어 벡터를 이용하여 다음 단어를 예측할 수 있고, DBOW는 문서의 벡터로 문맥을 예측할 수 있다[6]. Doc2Vec는 문단 내에 단어를 5개씩 순서대로 읽어드려 다음 단어를 예측하는 과정으로 문단 벡터가 학습하게 된다. 따라서 본 논문에서는 많은 문서를 문단 벡터 정보를 추출하기 위하여 Doc2Vec을 사용하였다. 김도우의 연구에서는 Doc2Vec가 같은 범주에 해당되는 문서에 대한 문서 벡터 표현을 생성하고 Word2Vec로 생성한 단어 벡터 표현들을 CNN에 적용하여 문서 분류 성능 향상을 위한 연구를 진행하였다[7].

2.3 DBSCAN

DBSAN은 Density model 중 하나이며 데이터의 위치 정보를 이용한다. 군집화에 많이 사용 되는 K-means는 군집 간 거리로 클러스터링을 하지만 DBSCAN은 밀도 기반의 클러스터링 방법으로 점이 세밀하게 밀집되어있어도 밀도가 높은 부분을 클러스터링 한다[8].

핵심 벡터의 최소 기준 거리 안에 인접 벡터의 개수가 일정 기준 이상이면 군집을 형성한다. 다음 Fig. 2은 DBSCAN 클러스터링의 예이다[9].

반경 내 점의 수(minPts)가 4라고 하면 핵심 벡터 P의 최소 기준 거리 내의 인접 벡터가 최소 점의 개수 이상일 경우 군집을 형성한다. P2의 경우는 최소 점의 개수가 3개이므로 군집의 중심이 되지는 못하지만 P의 군집에 속하기 때문에 이를 경계점이라고 한다. P4의 경우 어떠한 경우에도 범위에 포함되지 않는데 이를 노이즈라 한다. 즉, 핵심 벡터 P 중심으로 반경 내에 최소 점의 개수 이상의 점이 있으면

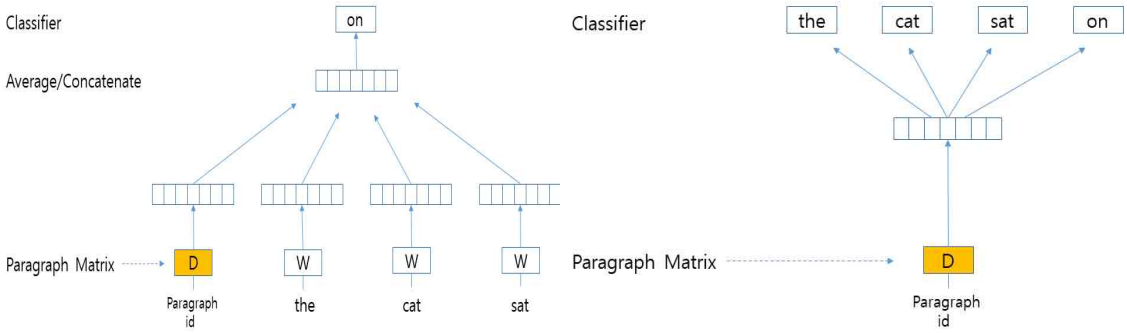

Fig. 1. Examples of DM and DBOW model configurations.

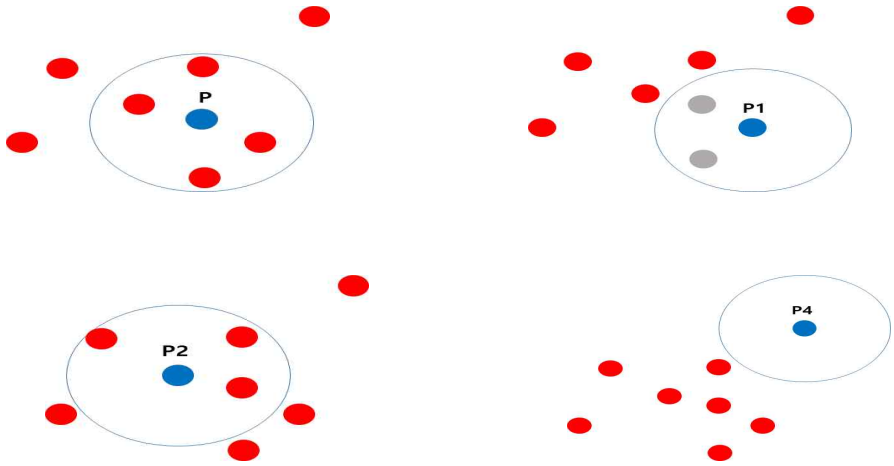

Fig. 2. Example of DBSAN Clustering.

P를 중심으로 군집을 형성하며, 다른 군집과 겹치면 그 군집을 서로 연결한다. 따라서 DBSCAN의 클러스터링은 다양한 형태의 군집이 형성된다[10]. 본 논문에서는 단어들의 유의어마다 다른 거리값을 추출하기 위하여 DBSCAN 알고리즘을 사용하였다.

3. 본 론

3.1 시스템 구성도

본 논문에서는 표절 검사 정확성의 향상을 위해 논문의 단락별 가중치를 부여한 TF-IDF 알고리즘을 사용하여 벡터값을 구하고 DBSCAN으로 유의어를 찾는다. 변환된 유의어와 초기 데이터 단어의 거리값과 L-R Table 명사 점수를 합하여 가중치 값을 적용 후 문장의 유사도를 측정한다. 다음 Fig. 3은 본 논문에서 제안하는 방법의 구성도이다.

실험에 사용된 데이터는 TF-IDF를 주제로 한 100편의 논문을 실험 데이터로 사용하였다. 데이터는 전처리 과정 후, 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 있고, 논문의 단락 별 중요도가 다르기 때문에 단락별 가중치를 적용한 TF-IDF 알고리즘을 적용한다. 이후 벡터값으로 R 기반 DBSCAN 알고리즘을 사용하여 단어별 군집화

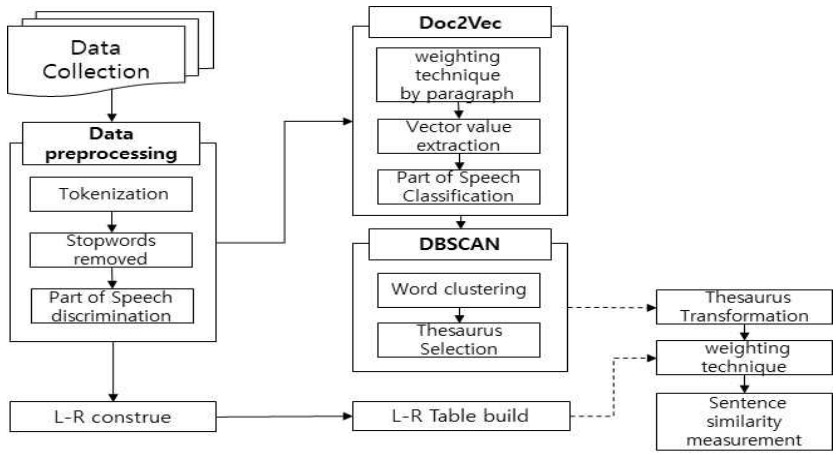

Fig. 3. System configuration diagram.

를 생성하고 유의어를 찾아 변환하고 L-R 구문 분석을 통하여 L-R Table을 구축한다. 변환된 유의어와 입력 문서 단어 간 거리값과 L-R Table 명사 점수를 합하여 가중치를 부여하고 문장의 유사도를 측정한다.

3.2 단락별 가중치 부여

선행 연구 결과를 통하여 논문의 단락별 단어의 빈도수가 다르다는 것을 알 수 있었다. 이는 서론에서 많이 출현하는 단어와 본론에서 많이 출현하는 단어가 다름을 알 수 있다. 예를 들어 서론과 관련 연구에서 A라는 단어의 알고리즘이 많이 등장하나 논문을 쓴 저자가 A알고리즘을 변형하여 A1이라는 알고리즘을 만들어 제시한 경우가 많았다. 이러한 경우 해당 논문에서 가장 중요한 단어는 A라는 단어가 아닌 A1이라는 단어가 중요함을 알 수 있다. 따라서 본 논문에서는 서론, 관련연구, 본론, 결론으로 이루어진 형식의 논문 단락에 가중치를 부여하는 방법을 적용한다. 서론은 1.5, 관련연구는 2, 본론은 4, 결론은 2.5의 가중치를 적용하였다.

3.3 벡터값 추출

단어 군집화를 진행하기 위해서 전처리 과정을 거친 데이터를 Python 기반 Gensim 패키지의 Doc2Vec 모듈을 사용하여 각 문서의 단락별 가중치를 부여하여 단어들을 빈도수가 높은 단어순으로 벡터값을 추출하였다. 다음 Table 1은 문서 A의 단락별 명사 단어 벡터값의 예이다.

문서 A의 전체 빈도수 중 가장 높은 단어는 TF-IDF이다. 단락별 가중치를 적용하였을 경우 Table 1과 같이 서론에서는 TF-IDF가 가장 높은 빈도수를 가졌으나 본론에서는 NTF라는 변형된 알고리즘이 높은 순위의 단어이다. 이처럼 단락별로 표절검사의 중요도가 높은 곳을 알 수 있다.

단락별 가중치를 적용하여 추출된 벡터값을 R기반 DBSCAN을 통하여 단어 군집화를 진행한다.

3.4 DBSCAN 군집화

다음 Fig. 4와 Fig. 5는 문서 50개의 군집화 결과이다. Fig. 4는 단락별 가중치를 적용하지 않고 진행한 군집이고 Fig. 5는 단락별 가중치를 적용하였다. 각 군집의 중심점에 위치한 단어를 대표 단어로 선정하였다.

대표 단어의 예로는 “수행” 이 선정 되었다. 군집에 형성된 단어들은 “수행하며”, “실행하다”, “수행하다” 등의 군집화가 형성되었다.

단락별 가중치를 적용하였을 경우 군집이 세분화가 되었다. Fig. 3에서 TF-IDF가 대표단어로 선정되었던 군집을 예로 보면 TF-IDF 와 BTF , NTF가 하나의 군집을 형성하였고 나머지 단어는 인접 군집으로 따로 형성되었다. 이는 단락별 가중치를 적용하였을 때 군집이 변하는 것을 확인 할 수 있다.

3.5 L-R 구문 분석

DBSCAN을 통해 측정한 유의어와 초기 문서 단어의 거리값은 수치가 낮아 유사도를 판별하는데 어려움이 있다. 본 논문에서는 유사도 판별의 정확성을 높이기 위해서 L-R 구문분석을 통하여 L-R Table을 구축하고 명사 점수를 유사도 판별하는 과정에 가중치로 적용하였다. 한국어의 구조 특성상 ‘L + R’ 구조로 ‘명사 + 조사’의 형태로 이루어지는데 조사들로 구성된 R의 명사가능 점수를 이용하여 L의 명사 점수를 알 수 있는데 이때의 명사 점수를 가중치로 적용하였다. R의 명사가능 점수는 세종말뭉치를 이용하여 학습된 명사가능 점수 데이터를 사용하였다. 다음 Table 2는 학습된 R데이터의 예이다.

Table 1. Example of vector value extraction by paragraph

Exordium		Related research		Main subject		Peroration	
Word	Vector value	Word	Vector value	Word	Vector value	Word	Vector value
TF-IDF	0.7643659	Algorithm	0.999024	NTF	0.743601	Keyword	0.864939
Document	0.6981274	TF-IDF	0.660862	TF-IDF	0.499234	NTF	0.708728
Weight	0.6954332	Keyword	0.563104	BTF	0.483869	Word	0.672409
Model	0.5113256	Text	0.484409	TF	0.301797	TF-IDF	0.574009
Review	0.4755161	Weight	0.47855	IDF	0.230176	List	0.564511

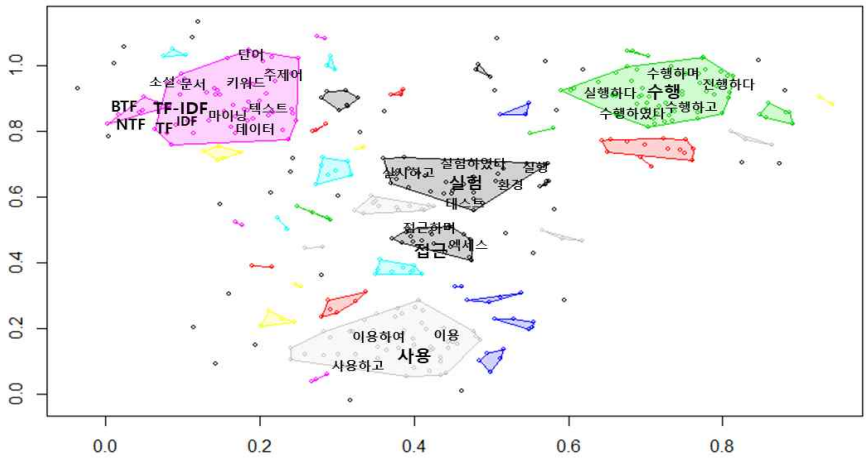

Fig. 4. Example of DBSCAN clustering (no weighting per paragraph).

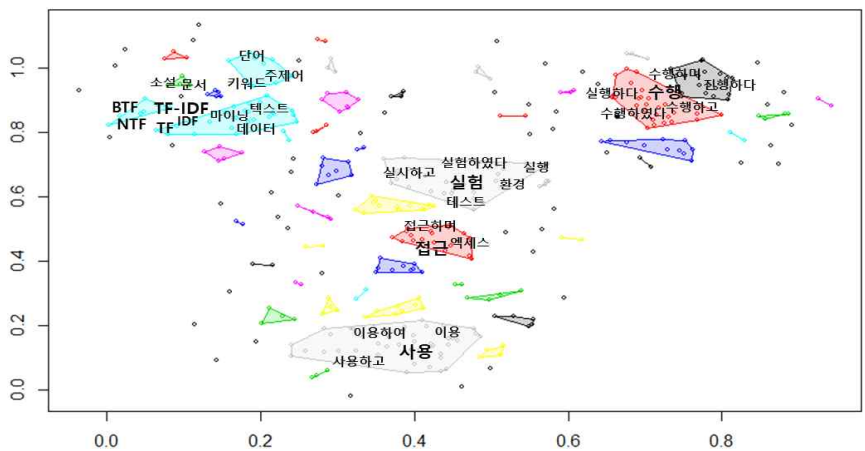

Fig. 5. Example of DBSCAN Clustering (Weighting by Paragraph).

R 데이터의 명사 가능 점수의 범위는 최소 -1, 최대 1의 범위에서 점수를 가지게 된다. 명사 가능 점수를 이용하여 명사점수를 계산은 다음 수식 3과 같다 [11].

$$S_L = \frac{(R_{N1} \times S_r) + \cdots + (R_{Nn} \times S_r)}{R_N} \quad (3)$$

Table 2. Examples of learned R data

Word	Possible noun score
했다	1.000000
하게	-0.926182
하며	0.327843
하는	0.079298
였다면	-1.000000

S_L은 명사 점수를 의미하고 R_{Nn}은 n번째 R의 등장 횟수 , S_r은 R의 점수 , R_N은 R의 총 등장 횟수를 의미한다. 만약 '적용+했다' 가 5번 , '적용+하며' 가 2번 등장하였다면 적용의 명사 점수는 수식 2와 같이 0.809의 점수를 가지게 된다.

$$\frac{(5 \times 1.0 + 2 \times 0.33)}{7} = 0.809 \quad (4)$$

4. 실험 결과 및 비교 분석

4.1 유사도 판별

다음 Table 3은 유의어 변형을 하지 않은 대표단어가 포함된 문서 A와 대표단어가 포함되지 않은 문서 B의 두 문장의 유사도를 계산한 예이다. 단어들의

Table 3. Sentence similarity evaluation of the documents A and B

Word vector	Similarity result
1,1,1,1,1,1,0,1,2,1,1,1,1,1,1,0,0,0,0	0.632189
1,1,0,1,2,0,0,0,1,1,0,1,0,1,0,1,1,1,1	

벡터 집합을 코사인 유사도 계산법으로 문장 유사도를 계산하였다[12].

유의어 변형을 하였을 때 유사도의 변화를 보기 위해 문서 B에 유의어 가중치를 적용한다. 유의어 변형을 적용한 문서 B를 문서 B1으로 단락별 가중치를 적용한 문서 B를 B2로 표기한다. 유의어에 대한 가중치는 '대표 단어와 유의어의 최대 거리값 - 해당 유의어 거리값'과 L-R Table에서의 S_L 점수를 합하여준다. 즉, '문장의 유사도 + (유의어 최대 거리값 - 해당 유의어 거리값)+ 명사 점수'를 합하여 준다. 명사 점수는 문장의 총 명사 단어 등장 횟수를 나누어준다. 따라서 문서 A, B1 문장의 유사도는 0.632 + (0.381 - 0.401) + 0.162 = 0.774이라는 유사도 값이 나오게 된다. Table 4는 단락별 가중치를 적용하지 않았을 때 문서 A와 B , 문서 A와 B1의 유사도 값과 단락별 가중치를 적용한 문서 A와 B1의 유사도 값을 나타낸다.

문서 A와 B의 문장 수는 각각 272 문장이다. 단락별 가중치를 적용 하지 않은 문서 A-B의 유사도의 합은 71.891의 결과 값이 나왔고 문서 A-B1의 유사도 합은 103.921의 값이 나왔다. 각 문서에는 유의어 변환에 대한 단어 간 거리값과 명사점수 값을 가중치로 부여 하였다. 하지만 가중치를 부여하였다고 문장 유사도 값이 단순히 높아지지 않고 2번 비교문장 유사도처럼 높은 문장 유사도 값에서 명사 점수 변동으로 인해 적은 문장 유사도 값이 나오는 것과 271번 비교문장처럼 높은 유사도 값으로 결과 값이 나오는 것을 확인 할 수 있다. 이는 유의어를 대표 단어로 변형 하여 거리값으로만 유사도 측정을 하는 방법보다 구문분석을 통해 유의어가 알맞은지를 비교 후 유사도를 측정하는 것이 더 좋은 정확도가 나옴을 알 수 있다. A-B2는 89.747이라는 유사도의 합이 나오는데 유의어 변환 가중치를 A-B1과 똑같이 적용하였음에도 A-B2가 더 낮은 유사도 값이 나옴을 알 수 있다.

문서 A-B의 유사성 문장을 예로 들면 A-B1 단어 군집시 '데이터' 와 '문서' 가 한 군집을 이루게 되기 때문에 유사도가 증가하게 된다. '알고리즘' 과 '변형식' 이라는 단어가 전체적 문장 구조에서 'TF-IDF' 단어 뒤에 나타나게 되어 유사하다는 결과가 나오지만 단락별 가중치를 적용하여 군집이 세분화 되어 한 군집에 있던 단어들이 서로 나누어지기 때문에 A-B2의 경우에는 유사도가 낮게 측정되는 결과가 나왔다.

4.2 비교 평가

비교 평가를 진행하기 이전에 문장 유사도의 기준을 선정한다. 유사도 기준은 실험 데이터의 총 문장 수의 문장별 유사도 평균값으로 선정하였다. 총 34,200 문장이며 문장별 유사도 합은 47832.1이다. 총 문장 수에서 유사도의 합을 나누어 0.715의 평균값이 나와 유사도의 기준을 0.7로 선정하였다.

Table 4. Noun score weighted similarity value

	No weights by paragraph		Weighting by paragraph
	Doc A-B	Doc A-B1	Doc 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
⋮	⋮	⋮	⋮
270	0.713	0.464	1.220
271	0.639	1.430	0.175
272	0.589	0.464	0.796
Sum	71.891	103.921	89.747

비교 평가는 LSA, LSA + N-gram, 제안 방법들의 결과들로 평가 하였다. LSA(Latent Semantic Analysis) 기반 유사도 측정과 본 논문에서 제안한 방법의 유사도 측정을 비교 분석한 결과이다. 구조 기반 방법은 텍스트 형식의 문서를 XML 형식의 문서로 변형하여 시퀀스 값의 평균 절대 오차차로 유사도 값을 측정하는 방법이고, LSA 기반 방법은 각 문장의 색인어를 벡터로 표현하여 유사도를 측정하는 방법이다. 문서는 유의어 변환을 하지 않은 두 문서와 유의어 변화를 진행한 두문서로 진행하였으며, 유사도가 높은 문장의 수를 측정하였다. LSA 와 N-gram을 함께 사용하여 유사도를 측정한 연구방법과 비교평가를 진행하였고 LSA와 N-gram의 임계값이 각각 0.7, 0.5의 경우의 유사도 값을 측정한다[13].

각 방법들의 문서 유사도 값과 정확률, 재현율, 조화평균을 통해 정확한 결과가 나왔는지를 평가하였다. 정확률, 재현율, 조화평균을 구하는 식은 다음 식 5와 같다.

$$Precisionratio(P)=\frac{tp}{tp+fp} \quad Recallratio(R)=\frac{tp}{tp+fn}$$

$$F-measure(F)=2\cdot\frac{P\cdot R}{P+R} \tag{5}$$

tp는 시스템이 유사한 문장이라고 제시한 문장 중에 맞는 문장 수이고, fp 는 유사한 문장이라고 제시한 문장 중에 틀린 문장의 수이며, fn 은 유사한 문장이 아니라고 제시한 문장 중에 유사한 문장의 수이다. 다음 Table 6은 각 비교 평가의 문서 유사도 및 정확률, 재현율, 조화평균의 결과 값이다.

문서 유사도는 LSA + N-gram 의 경우 31.1 %로 가장 높고 제안한 방법은 29.4 %가 나왔다. 하지만 정확률 조화평균은 제안 방법의 경우 높은 결과가 나왔다. 이는 본 논문에서 제안한 방법의 비슷한 단어이지만 다른 의미를 지니는 단어들에 대해 정확한 유사도 측정 결과가 나오는 것을 알 수 있다.

5. 결 론

본 논문에서는 논문 표절 검사 정확성을 향상시키기 위해 단어들을 군집화 하여 유의어를 찾아 대표 단어를 선정하고 L-R 구문분석을 통해 가중치를 부여하여 문장의 유사도를 판별하였다. 유의어를 찾기 위해 전처리 과정을 거친 문서를 단락별 가중치를 적용한 TF-IDF 알고리즘과 Doc2Vec로 벡터화 하여 DBSCAN으로 단어 군집화를 진행하였다. L-R구문분석으로 L-R table을 구축하고 R의 명사 가능 점수로 L 명사 점수를 구하여 문장 유사도 판별에 가중치로 적용하였다. 유사도는 유의어 변형 가중치와 단락별 가중치 적용하지 않은 문서 A-B 유의어 변형 가중치를 적용한 문서 A-B1, 유의어 변형 가중치와 단락별 가중치를 적용한 A-B2의 유사도를 비교하였다. 그 결과 유의어 변환 가중치를 적용하였을 때 높은 유사도 점수가 나오나 문장별 유사도를 비교해 보면 유의어 변형을 진행 하였어도 L-R구문분석을 통한 명사점수를 통해 구문 형태가 맞지 않아 낮은 유사

Table 5. Examples of similar sentences

Doc A	Doc B
⋮ <표 1>은 실험 데이터를 TF-IDF 알고리즘에 적용하여 얻은 결과를 나타낸다. ⋮	⋮ 표 3은 뉴스 문서를 대상으로 TF-IDF 변형식을 적용한 결과이다. ⋮

Table 6. Comparison of similarity measure

Document	Analysis		
	LSA method	LSA + N-gram method	Suggested method
Document similarity	15.9 %	31.1 %	29.4 %
Precision	58.7 %	83.9 %	94.1 %
Recall	48.2 %	95.7 %	91.6 %
F-Measure	0.52934	0.89412	0.92833

도가 나오는 것을 알 수 있었고 단락별 가중치를 적용하였을 때 본론에서 변형된 알고리즘을 의미하는 단어를 사용하였을 경우에는 유사도 값이 낮아지는 것을 알 수 있었다. 이는 전체 문서를 가지고 유사도를 비교 하는 것보다 단락별 가중치를 통하여 문서의 유사도를 비교하는 방법이 더 정확함을 알 수 있다.

REFERENCE

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

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

[3] R. Robertson, "Understanding Inverse Document Frequency: on Theoretical Arguments for IDF," *Journal of Documentation*, Vol. 60, No. 5, pp. 503-520, 2004.

[4] J.Y. Son and Y.T. Shin, "Music Lyrics Summarization Method Using TextRank Algorithm," *Journal of Korea Multimedia Society,* Vol. 21, No. 1, pp. 45-50, 2018.

[5] Q. Le and T. Milokov, "Distributed Representations of Sentences and Documents," *Proceeding of the 31st International Conference on Machine Learning*, Vol. 23, No. 12, pp. 698-702, 2014.

[6] K. Cheng, J. Li, J. Tang, and H. Liu, "Unsupervised Sentiment Analysis with Signed Social Networks," *Proceeding of the 23rd ACM Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discorvery and data Mining*, pp. 777-786, 2017.

[7] D.W. Kim and M.W. Koo, "Categorization of Korean News Articles Based on Convolutional Neural Network Using Doc2Vec and Word2Vec," *Journal of Korea Institute on Information Scientists Engineers*, Vol. 44, No. 7, pp. 742-747, 2017.

[8] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, USA, 2005.

[9] M.S. Kwon, Y.H. Kang, H.J. Han, and D.S. Cho, "Adaptive DBSCAN for Time-varing Clustering DBSCAN," *Proceeding of Information and Control Symposium*, Vol. 2016, No. 4, pp. 134-135, 2016.

[10] M. Ester, H.P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceeding of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[11] Y.H. Won, *Efficient LR(k) Parsing Algorithms*, Master's Thesis of Korea Advanced Institute of Science, 1975.

[12] M.J. Kim and S.J. Lee. "Measures of Abnormal User Activities in Online Comments Based on Cosine Similarity," *Journal of the Korea Institute of Information Security and Cryptology*, Vol. 24, No. 2, pp. 335-343, 2014.

[13] H.S. Ji, J.H. Joh, and H.S. 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.

김 병 식

2016년 조선대학교 전기공학과 공학사
2016년~현재 조선대학교 소프트웨어융합공학과 석사 과정
관심분야 : 자연어 처리, 오피니언 마이닝, 기계학습

신 주 현

1986년~2011년 ㈜청전정보 팀장, ㈜투루텍 기술이사
2007년 조선대학교 전자계산학과 이학박사
2018년 조선대학교 미래사회융합대학 ICT융합학부 부교수
관심분야 : 멀티미디어 데이터베이스, 빅 데이터 분석, 텍스트마이닝, 감성정보 처리 등