문서 유사도 분석 도구에 관한 연구

김희진*, 강홍비*, 김한성*
*한신대학교 컴퓨터공학부
e-mail: erth1016@hs.ac.kr

A Study on Tools for Text Similarity Evaluation

Hong-Bi Kang*, Hee-Jin Kim*, Han-Sung Kim*
*Dept of Computer Science, Han-Shin University

요 약

본 시스템은 LSA 또는 벡터공간 모델 방식을 이용하여, 문장 대 문장, 문서 대 문장, 다중 문서 간 유사도 분석을 수행한다. 이는 문서의 특수문자를 제거한 뒤, 형태소 분석을 기반으로 단어를 추출하여 TF-IDF 가중치를 추출한뒤 행렬 계산을 통하여 Cosine 계산식을 사용하여 유사성을 검출하는 단계로 구성된다. 제시된 기법은 2개의 오픈소스를 이용하며, x86 기반 64bit Windows에서 개발되었으며, 60% 이상의 정확도를 나타낸다.

1. 서론

최근 사회가 빠르게 정보화됨에 따라, 표절로 인한 지적 재산권 침해에 대한 분쟁의 빈도가 증가하고 있다. 유명 인사들의 표절 문제를 겪으며, 표절은 단순히 금전적 손실을 끼치는 것이 아니라 국가 이미지까지 저하시키고 있다. 이에, 지적 재산권을 보호하기 위해 문서 유사도를 탐지할 수 있는 기술이 요구되고 있다.

어순이 일정한 외국어와 달리, 한국어는 어순이 유동적이기 때문에 표절을 탐지하기 어렵다. 이러한 문제를 해결하기 위해 본 논문은 자연어 처리기술을 이용한 문서 유사도 분석 도구를 제시하고자 한다. 본 연구는 두 개의 문서 또는 텍스트뿐만 아니라 여러 개의 문서 간 유사도 까지 확인할 수 있게 하는 것을 목적으로 한다.

본 연구는 동적 분석 프로세스를 사용 하는데, 이는 악의적으로 문장의 재배치 혹은 재구성을 이용하여 표절을 은폐하려고 시도하여도 의미 인식 분석을 통해 속임수를 예방한다. 제안된 기법은 2개의 잘 알려진 오픈소스를 이용하여 유사도 분석 및 표절감시가 실시된 x86기반 64bit Windows 환경에서 시험되었으며, 대체로 60% 이상의 정확도를 나타낸다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개한 뒤, 3장에서는 문서 유사도 분석 도구의 전체 구조에 대해 설명한다. 3장에서는 실험 결과를 4장에서는 결론 및 향후 연구 방법을 논한다.

2. 연구 방법 및 이론

1) 연구 방법

본 연구는 형태소 분석을 기반으로 문서 유사도 분석 도구를 설계, 구현하여 한국어로 구성된 문서의 유사도를 분석하는 것을 내용으로 한다. 구체적인 연구의 방법은 다음과 같다.

첫째, 분석하려는 문서의 특수문자를 모두 제거한다.

둘째, 형태소 분석을 통하여 각 문서별 어절을 추출한다.

셋째, 추출된 형태소를 기반으로 2개 이상의 표절 탐지 기법을 선택하여 유사도를 분석한다.

2) 이론적 배경

가. 형태소 분석

“의미가 있는 언어의 형태론적 최소단위”를 형태소라고 한다. 한국어는 다양한 대명사, 조사 등을 가지고 있어 하나의 독립적인 어절을 형태소로 판단해야 한다.

나. 유사도

(1) 표절 유형

<표 1>에서 볼 수 있듯 한국어는 대표적으로 3가지의 표절 유형을 가지고 있다. 표절 유형마다 사람이 쉽게 표절을 알아볼 수도 있으나, 쉽게 알아보지 못하는 경우도 있다. 각 유형마다 특징이 뚜렷하기 때문에, 이에 대한 유사도를 검출하기 위해 다양한 표절 탐지 기법들이 연구되었다.

<표 1> 유형별 표절 예제

표절 유형	표절 예제 문장
단어 치환	지난 6.10 범국민대회 참가자들에게 방패를 휘두른 의경 2명에 대해 경찰이 징계 절차에 착수했습니다.
	지난 6.10 범국민대회 참여자들에게 방패로 공격한 의경 2명에 대해 경찰이 징계 절차를 시작했습니다.
어순 변경	음주와 비만, 운동부족 등도 장암의 원인으로 거론되고 있지만, 적색육 및 가공육의 섭취가 특히 위험한 발병 요인이라는 과학적 증거가 최근 부쩍 많이 제시되고 있다.
	적색육 및 가공육의 섭취가 비만, 운동부족등이 원인이 되는 장암의 발병 요인이라는 과학적 증거가 최근 부쩍 많이 제시되고 있다.
원문 요약	계산을 하거나 하지 않는 것도 필요하지만 중요한 사실은 외부의 명령어를 읽어 그것을 수행하는 하나의 완결적인 제어 메커니즘을 갖게 되었다는 점이다
	계산도 필요하지만 중요한 사실은 명령어를 읽어 수행하는 제어 메커니즘을 갖게 되었다는 점이다

(2) 표절 탐지 기법

표절 탐지 기법은 <표 2>에서 볼 수 있듯이 크게 6가지로 나뉜다. 가장 흔히 사용되는 표절 탐지 기법은 벡터 공간 모델 방식인데, 이는 원본 문서와 비교하고자 하는 문서의 키워드가 정확히 일치해야 한다는 단점이 있다. 따라서 본 연구에서는 정확도 향상을 위해 단어들의 의미 관계를 기반으로 한 유사 여부 판별 방식인 LSA 방식을 혼합하여 사용하였다.

<표 2> 표절 탐지 기법

표절 탐지 기법	방법	특징
어절 트리 비교	빈도수 어절 인덱스를 이용해 인덱스 간의 유사도를 비교	-단어 빈도의 유사도에 의존
패턴 매칭 방식	프로그램 소스를 패턴화시켜 하나의 언어로 매칭하여 탐지	-탐지 시 조건이 제한적
지문법 이용	문서의 단어들의 유사성, 단어의 빈도수를 비교하여 통계적으로 분석	-유사어 사용, 문법을 변경할 경우 탐지 불가 -문서량 증가시 시간 많이 소요
N-gram 방식	N개의 음절을 추출해내, 그것들의 일치도를 비교	-많은 저장 공간이 필요 -전혀 다른 문장도 유사한 것으로 판정 가능
벡터 공간 모델 방식	문장에서 색인어를 추출하여 공간상의 벡터로 표현해서 유사도를 계산	-추출되는 키워드가 정확히 일치해야 함
LSA 방식	단어들의 의미 관계를 찾아 유사 여부 판별	-정확도가 떨어짐

(3) TF-IDF

TF(Term Frequency, 단어빈도)는 특정 단어가 문서 내에 얼마나 자주 등장하는지를 나타내는 값으로 이 값이 높을수록 문서 내에서 중요한 단어라고 볼 수 있다.

DF(Document Frequency, 문서빈도)는 전체 문서 중 해당 문서가 출현한 문서의 개수인데, 이 값의 역수를 IDF라고 하며, IDF 값은 문서군의 성격에 따라 결정된다.

TF-IDF(Term Frequency - Inverses Document Frequency)는 여러 문서로 이루어진 문서군에서 어떤 단어가 특정문서에서 중요한 것인지를 나타내는 통계적 수치이다.

<그림 1> TF-IDF 계산식

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}} \qquad \mathrm{idf}(t,D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

(4) LSA

다변량 통계분석 방법으로, 고차원 데이터 공간에 대해 축을 변경하여 데이터에 내재해 있는 구조를 밝히는 기법이다. 축은 SVD(Singular Value Decomposition) 라는 통계적 기법을 사용하여 찾는다.

LSA를 사용하기 위해 문서마다 단어들의 빈도를 행렬로 생성하여, SVD 공식 $A_{t\times d} = T_{t\times n}S_{n\times n}(D_{d\times n})^T$에 대입한다. T는 단어, D는 문서에 대응하는 행렬이다. 해당 공식을 이용하여 차원 축소를 한다.

<그림 2>는 LSA 뿐만 아니라 벡터 공간 모델 방식 등 여러 가지 표절 탐지 기법에 이용된다.

<그림 2> Cosine 유사도 계산식

$$S_c = \mathrm{sim}(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\,\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

3. 문서 유사도 분석 도구 전체 구조

1)문서 유사도 분석 기법의 단계

문서의 유사도를 분석하기 위해서는 크게 6개의 단계가 필요하다.

첫째, 분석하려는 문서 내용의 특수 문자 제거

둘째, 특수문자가 제거된 내용을 형태소 추출

셋째, 형태소로 추출된 내용을 문서별로 TF-IDF 가중치 추출

넷째, 추출된 TF-IDF 이용하여 행렬 생성

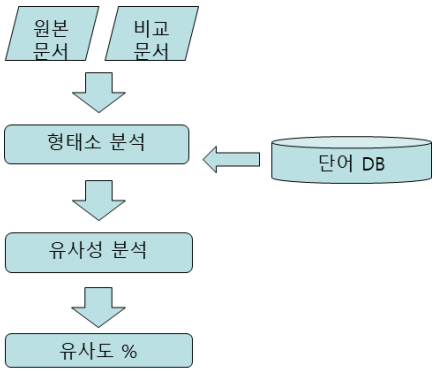
다섯째, 생성된 행렬에 LSA 기법 적용 후 S*Dt 행렬 생성

여섯째, 생성된 행렬에 Cosine 계산식 사용하여 유사성 검출

2) 문서 유사성 분석 구성도

본 연구에서 개발한 문서 유사성 분석 도구는 <그림 3> 과 같은 시스템의 흐름을 보여준다. 원본 문서와 비교 문서를 문서 유사도 분석 도구에 입력하면 형태소 분석기에서 문서를 형태소로 분석하여 유사성 분석기를 이용하여 유사도 %를 추출한다. 형태소를 분석할 때, 단어 DB의 사전을 바탕으로 형태소를 어절 단위로 나누어 준다.

<그림 3> 문서 유사성 분석 구성도



3) 형태소 분석 및 구문 분석

형태소 분석 및 구문 분석은 <표 3>의 과정을 통해 수행된다. 우선, 띄어쓰기를 기준으로 토큰화 한 뒤, 색인어 분석기를 이용하여 토큰에서 색인어를 추출한다.

<표 3> 형태소 분석 및 구문 분석

형태소 분석 및 구문 분석

```
Public ArrayList<String> extractionNoun(String searchQuery) throws MorphException{
ArrayList<String> nounList = new ArrayList<String>();
MorphAnalyzer maAnal = new MorphAnalyzer(); //형태소 분석기
StringTokenizer stok = new StringTokenizer(searchQuery," "); //쿼리문 띄어쓰기 기준으로 토큰화
//색인어 분석기를 통해 토큰에서 색인어 추출
while(stok.hasMoreTokens()){
    String token = stok.nextToken();
    //형태소 분석
    List<AnalysisOutput> indexList = maAnal.analyze(token);
    for(AnalysisOutput morpheme : indexList)
    //명사추출
    if(morpheme.gerPos()=='N')
    nounList.add(morpheme.getStem());
    }
}
```

4) 유사도 분석

형태소 분석 결과를 바탕으로 TF-IDF , SVD 수식을 이용하여 값을 계산한 뒤 Cosine Vector를 계산하여 문서 유사도를 검출한다.

<표 4> Cosine 벡터 계산

Cosine 벡터 계산

```
#코사인 벡터 계산
def vector(aA,bB):

    A = aA
    B = bB
    mA = mat(A)
    mB = mat(B)

    np.seterr(divide='ignore', invalid='ignore')

    cossim_AB = dot(mA, mB.T) / (linalg.norm(mA) * linalg.norm(mB))

    if cossim_AB < 0 or math.isnan(cossim_AB):
        cossim_AB = 0

    if aA[0] == bB[0] and aA[1] == bB[1]:
        cossim_AB = 1

    print str(int(cossim_AB*100));
```

4. 실험 결과

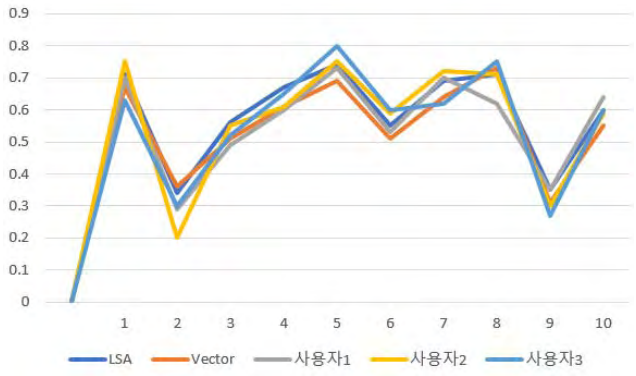
본 연구에서 개발한 문서 유사성 분석 도구를 사용하여 10쌍(20개)의 원본문서와 대조문서를 이용하여 문서 유사도를 분석한 결과는 <표 5>와 같다.

<표 5> 문서 유사도 분석 결과

	LSA	Vector	사용자1	사용자2	사용자3
1	71%	67%	70%	75%	63%
2	34%	36%	29%	20%	30%
3	56%	51%	49%	55%	52%
4	67%	61%	60%	61%	65%
5	74%	69%	73%	75%	80%
6	55%	51%	53%	59%	60%
7	69%	64%	70%	72%	62%
8	71%	73%	62%	71%	75%
9	35%	31%	35%	30%	27%
10	60%	55%	64%	59%	60%

한 쌍의 문서를 LSA 방식과 벡터공간모델 방식으로 비교했을 경우 10% 이내의 오차가 발생하였다. 이는 문서마다 달랐는데, 그 이유는 문서마다 구성이 다름에서 찾을 수 있었다.

<그림 3> 문서 유사도 분석 그래프



<그림 3>은 사용자1, 사용자2, 사용자3의 유사도 판단과 LSA 와 벡터공간모델방식으로 계산된 유사도를 비교한 그래프이다. 위 그래프를 통해 사람의 판단이 대개 프로그램을 통해 계산된 유사도보다 높음을 알 수 있으며, 오차범위 또한 10% 내외로 비교적 정확한 편임을 알 수 있다.

5. 결론

본 논문에서는 LSA방식과 벡터공간모델 방식을 기반으로한 문서 유사도 분석 도구를 제안하였다. 제안된 도구는 문서의 유사도 분석을 위해 크게 세 단계로 나뉜다. 문서 파일에 대한 형태소 분석 단계를 통해 형태소를 분석한 뒤, 해당 결과에 대해 유사성 분석을 진행한 뒤, 이를 시각화 하여 사용자에게 제공한다.

첫 번째 단계인 형태소 분석 단계에서는 문서의 특수문자를 제거하고, 문서의 형태소를 분석한 뒤, 결과를 저장하고, 추출된 형태소를 TF-IDF 가중치를 적용하는 단계이다. 이를 통해 어절 단위의 한국어의 형태소를 효율적으로 분석할 수 있도록 하였다.

두 번째 단계인 유사성 분석 단계에서는 생성된 행렬에 SVD 공식을 적용한 뒤 S*Dt를 이용한 행렬을 생성하고, 생성된 행렬을 Cosine 계산식을 이용하여 유사도를 검출하는 단계로, LSA 와 벡터공간모델방식을 효율적으로 사용하여 정밀도를 높였다.

마지막으로 유사도 분석 결과를 사용자에게 시각화하여 전달함으로서 문서 유사도를 쉽게 확인할 수 있도록 하였다.

LSA와 벡터공간모델을 활용하는 문서 유사도 분석 도구는 기존의 유사도 분석 도구와 비교할 때 다음과 같은 장점을 확인할 수 있었다.

- 어순이 불규칙한 한국어의 형태소를 정확하게 분석하여 문서 유사도 분석에 있어서 정확도를 높였다.
- 유사도 분석 과정에서 LSA방식과 벡터공간모델방식을 사용하여 정확도를 높였다.
- 시각화를 통해 사용자가 문서 유사도 분석 결과를 한눈에 확인할 수 있다.

또한, 20여개의 문서 비교 실험을 통하여 문서 유사도 분석 도구를 사용하였을 때와 사람이 유사도를 분석하였을 때의 결과가 유사함을 실험을 통해 확인하였다.

참고문헌

[1] 이수한 (2013) "문헌의 문장 유사도 분석 모델에 관한 실증적 연구"
[2] 임진수 (2012) "코딩 스타일을 고려한 코드 표절 검출 시스템"
[3] 임나리 (2011) "실시간 리포트 표절 검사를 위한 임계 알고리즘 개발 연구"
[4] 조준희 (2010) "한국어 문서 표절 검사를 위한 LSA와 N-gram 기반의 유사 문장 판별 방법"
[5] 박희완 (2010) "자바의 정적 트레이스 버스마크를 통한 소프트웨어 도용 탐지"
[6] 박주열 (2009) "문서 구조 정보를 이용한 유사도 검사 시스템의 설계 및 구현"
[7] 한소정 (2009) "오픈 소스코드 표절 탐지 기법"
[8] 감태성 (2007) "중간표현과 서열정렬 알고리즘을 이용한 소스코드 표절 탐지 시스템에 관한 연구"
[9] 문승미 (2007) "계층적 응집 클러스터링 기법을 이용한 소스 코드 표절 검사"
[10] 진명재 (2005) "대용량 한글 문서를 위한 표절 검색 시스템 개발"
[11] 유영균 (2005) "표절 프로그램 과제물 검사 시스템 개발"
[12] 양승진 (2004) "유전알고리즘을 이용한 프로그램 표절 검사"
[13] 김용건 (2004) "구조 비교를 통한 프로그램 표절 검사"
[14] 천승환, 김미영, 이귀상 (2008) "유사어절트리와 비색인어기반의 문서표절 유사도 분류법"
[15] 지혜성, 조준희, 임희석 (2010) "한국어 문장표절 유형을 고려한 유사문장판"
[16] 안병렬, 김문현 (2006) "효율적인 프로그램 표절탐지에 관한 연구"

-본 논문은 2017년 한이음 ICT멘토링 프로젝트의 결과물입니다.-