J. Inf. Technol. Appl. Manag. 24(1): 1~10, March 2017
https://doi.org/10.21219/jitam.2017.24.1.001

ISSN 1598-6284 (Print)
ISSN 2508-1209 (Online)

표절 탐지를 위한 비트 시그니처 기법

김우생* · 강규철**

Big Signature Method for Plagiarism Detection

Woosaeng Kim* · Kyucheol Kang**

Abstract

Recently, the problem of plagiarism has emerged as a big social issue because not only literature but also thesis become the target of plagiarism. Even the government requires conformation for plagiarism of high-ranking official's thesis as a standard of their ethical morality. Plagiarism is not just direct copy but also paraphrasing, rewording, adapting parts, missing references or wrong citations. This makes the problem more difficult to handle adequately. We propose a plagiarism detection scheme called a bit signature in which each unique word of document is represented by 0 or 1. The bit signature scheme can find the similar documents by comparing their absolute and relative bit signatures. Experiments show that a bit signature scheme produces better performance for document copy detection than existing similar schemes.

Keywords : Plagiarism, Bit Signature

Received : 2017. 01. 03. Revised : 2017. 03. 24. Final Acceptance : 2017. 03. 24.

※ The present Research has been conducted by the Research Grant of Kwangwoon University in 2016.

* Corresponding Author, Professor, Department of Computer Software, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01897, Tel : +82-(0)2-940-5114, e-mail : kwsrain@gmail.com

** Department of Computer Software, Kwangwoon University, e-mail : pwww1111@naver.com

1. 서 론

최근 저작권에 대한 관심과 중요성이 높아짐에 따라 문서 표절 탐지에 대한 필요성이 증대되고 있다. 표절(剽竊)이란 다른 사람이 쓴 문학 작품이나 학술 논문, 또는 기타 각종 글의 일부를 직접 베끼거나 아니면 관념을 모방하면서, 마치 자신의 독창적인 산물인 것처럼 공표하는 행위를 가리킨다.[1] 문서를 표절하는 방법은 다양한데, 원본의 내용을 일부 또는 그대로 복사하는 방법 뿐 아니라, 원본에서 사용한 내용과 유사한 동의어를 사용하여 표절하는 방법, 부분적인 내용을 삭제하거나 순서를 바꾸는 방법, 문서나 문장의 구조를 바꾸는 방법 등이 있다.

어문 저작물의 표절은 어린 학생들의 숙제부터 나아가 대학 수시 모집에서의 자기소개서 및 학위 논문에 이르기까지 광범위하게 이루어지고 있다. 최근 고위 공직자의 논문이 표절로 밝혀지면서 사회적인 논란이 일고 있다. 윤성규 환경부 장관[2013] 서남수 교육부 장관[2014], 홍영표 통일부 장관[2015], 정진엽 복지부 장관[2015], 김병준 총리[2016] 등은 표절 의혹을 받았거나 이로 인해 후보직에서 낙마한 경우들이다. 이에 민주당 이상민 의원은 고위공직자에게는 높은 도덕성이 요구되는 만큼 객관적이고 공정한 절차로 논문 표절 여부를 검증해야 한다며 인사청문회법 일부 개정 법률안을 대표 발의했다. 인사청문회 대상 공직자 임명 시 사전에 논문표절 검증을 의무화하는 것이 골자다. 개정안은 임명동의안 등 공직 후보자의 임명 관련 첨부서류에 논문 표절 여부 심사와 관련한 사항이 들어가도록 했다.

문서의 표절 여부는 사람이 가장 잘 판단할 수 있지만 많은 양의 정보를 일일이 검사하기는 사실상 불가능하기 때문에, 자동으로 문서의 유사도를 검색하여 표절 여부를 판단하는 시스템에 관련 연구와 제품이 개발되고 있다. 표절 검사를 하는 방법에는 크게 대상에 따라 문서 표절 검사와 프로그램 소스 코드 표절 검사가 있다. 프로그램 소스 코드는 문서에 쓰이는 자연어에 비해 형식에 제약이 있고 의미를 나타내는 단어의 개수가 크지 않기 때문에 많은 연구가 진행되어 있는 반면, 문서 표절 검사 방식은 자연어의 복잡성으로 말미암아 아직 많은 연구가 필요한 분야이다[Kim and Cho, 2008].

본 논문은 문서 표절 검사에 관한 것으로 특히 주어진 문서와 가장 유사도가 높은 비교 대상 문서들을 효율적으로 그리고 정확히 순위화 하는 방법을 제안한다. 예를 들어, 어떤 공직자의 학위 논문의 표절이 의심이 된다면, 많은 논문들 중에서 유사도가 가장 높은 순서로 논문들을 순위화 하는 방법이 필요하다. 이를 위해 본 논문에서는 각 문서의 고유한 단어를 0 또는 1의 비트로 표현하는 비트 시그니처 기법을 제안한다. 두 문서 간의 유사도는 두 문서의 비트 시그니처 간의 절대적인 겹침(Absolute overlap) 즉, 주어진 문서를 기준으로 두 문서 간의 겹칩을 조사하는 방법과, 상대적인 겹침(Relative overlap) 즉, 사이즈가 더 큰 문서를 기준으로 두 문서 간의 겹칩을 조사하는 방법도 함께 고려한다. 비트 시그니처 기법은 문서의 단어순서와 무관하게 단어들을 표현하기 때문에 문서의 구조 변경이 성능에 영향을 미치지 않고, 문서 간의 유사도 비교 뿐 아니라 문장 안의 유일한 단어를 비트 시그니처 기법의 비트로 표현하여 문장 간의 유사도 비교에서도 사용할 수 있다는 장점이 있다.

본 논문은 다음과 같이 구성된다. 제 2장은 문서의 표절과 관련된 연구에 대해서 소개하며 제 3장은 문서 간의 겹침을 측정하여 서로 다른 문서 간의 표절을 찾는 기존의 대표적인 기법들과 본 논문에서 제안하는 비트 시그니처 기법을 설

1) https://ko.wikipedia.org/wiki/%ED%91%9C%EC%A0%88.

명한다. 제 4장은 실험을 통하여 제안하는 비트 시그니처 기법이 기존의 기법들보다 성능이 우수함을 보인다. 마지막으로 제 5장에서 결론과 향후 연구 과제에 대해서 언급한다.

2. 관련 연구

문서 표절 검사 방식에는 크게 문맥적 의미를 이용한 방식과 문장 구조를 이용한 방식이 존재한다. 이 중 문맥적 의미를 이용한 방법은 문장의 구조나 단어의 순서에 상관없이 문장의 뜻을 이용하여 표절 탐지를 하기 때문에 가장 이상적인 표절 검사 방식이나, 자연어의 의미를 파악하는 작업이 어려우며 속도 또한 느리다는 단점이 있어 많이 사용되지는 않는다. 반면 문장 구조를 이용하여 표절 탐지를 수행하는 방법은 현재 문서 표절 검사 방식에서 활발히 연구되고 있는 분야로 크게 속성 계수법(Attribute counting) 방식과 구조적 검사(Structure metric) 방식이 존재한다[Donaldson et al., 1981].

속성 계수법 방식은 문서에서 자주 사용되는 단어들 간의 유사성이나 빈도를 검사하여 표절 정도를 측정하는 방식이다. 단어들 간의 유사성이나 빈도에 초점을 맞추기 때문에 표절 탐지 시간이 문서의 길이에 크게 영향을 받지 않으며 문서의 단락 순서가 바뀌어도 성능에 영향을 미치지 않는다. 하지만 부분 표절 탐지가 어렵다는 단점이 있다. 이 방식의 대표적인 기법인 벡터 공간 모델 방식은 정보 검색(Information Retrieval) 모델의 한 종류로써, 각 문서의 단어를 가중치가 부여된 벡터로 표현하여 문서 간의 유사성을 측정하는 방법이다[Salton, 1992]. 반면 SCAM(Stanford Copy Analysis Mechanism) 기법은 문서에 포함 된 단어들의 상대적 발생 빈도와 문서 간의 포함 관계를 통해 유사도를 찾는다[Shivakumar and Garcia-Molina, 1995]. 특히 이 기법은 두 비교 문서에 겹치는 단어의 발생 빈도에 따라 다른 유사도 값을 갖도록 하였다. 또한 프로그램 유사성 검사기(CloneChecker) 기법은 프로그램 간의 표절을 탐색하기 위해 프로그램 요약 단계, 프로그램 간 유사성 비교 단계, 유사한 것끼리 그룹 짓는 단계를 거친다[Jang et al., 2001]. 이 기법에서는 유사성을 비교하기 위해 두 프로그램의 구문 트리 전체에서 공통된 구문 트리들이 차지하는 비율을 계산하였다.

구조적 검사 방식은 문서에 포함된 단어의 정확한 일치가 아닌 토큰 스트링(Token String)의 유사성을 계산하여 표절 탐색을 하는 방법이다. 표절 탐색 시간이 문서의 길이에 영향을 받으며 문서의 단락 순서 또한 표절 탐색 성능에 영향을 미친다. 반면 부분 표절 탐지가 용이하며, 프로그램 소스 코드를 검사할 때 더 효과적으로 사용된다. 대표적인 SIM(A Utility For Detecting Similarity in Computer Programs) 방법은 두 프로그램 간의 표절을 탐지하기 위해 먼저 프로그램을 어휘 분석기를 통해 파스 트리로 변환한다. 파스 트리를 연속되는 여러 개의 토큰으로 구성되는 스트링으로 보고, 다시 스트링을 비교 단위로 하여 두 프로그램의 유사성을 비교한다[Gitchell and Tran, 1999]. YAP(Yet Another Plague) 방법은 스트링 매칭 방법을 사용하여 문서를 토큰 시퀀스로 변경시키는 단계와 토큰 스트링을 서로 비교하는 단계를 가진다[Wise, 1996]. 특히 이 방식은 가능한 최대 공통의 연속적인 부분 스트링을 찾기 위한 새로운 Running-Karp-Rabin Greedy-String-Tiling 알고리즘을 제안하였다. 또한 하드 매칭 기법에 비해 단어의 도치, 누락, 오타에 뛰어난 성능을 보인 집합 기반 POI(Point of Interest) 검색 알고리즘과[Go and Lee, 2013], 이 기법의 데이터 로딩 알고리즘과 텍스트 검색 알고리즘을 변형하고 어절 연산 알고리즘을 추가하여 문서 유사도 측정에 응용한 기법도 있다[Go and Lee, 2014].

한편 속성 계수법 방식과 구조적 검사 방식을 같이 사용하는 기법들도 있는데, DEVAC(Document EVolution Analyzing Center)의 경우 두 번의 표절 탐색을 수행하는데 처음에는 빠른 검색을 위해 문서의 길이에 크게 영향을 받지 않는 속성 계수법 방식 중 하나인 fingerprint[Schleimer et al., 2003]를 이용하여 표절 탐색 문서 군의 크기를 줄인 다음, 구조적 검사 방식인 지역 탐색을 통해 자세한 표절 탐색을 수행한다[Ryu et al., 2008]. 벡터 공간 모델 기법은 색인어로 추출되는 단어가 정확히 일치해야 한다는 문제가 있다. 이러한 문제를 해결하기 위해 국소 문맥 정보를 이용해서 단어들 간의 의미 관계를 찾아 벡터로 표현하는 LSA(Latent Semantic Analysis) 모델과 두 문장 내에 존재하는 인접한 N개의 음절인 N-gram을 추출하고 그것들 중에서 얼마나 많은 N-gram이 일치하는지를 함께 고려하여 문서의 유사 여부를 판단하는 기법도 제안되었다[Ji, H. et al., 2010].

3. 겹침 측정 방법(Overlap Measure Method)

본 장에서는 문서 표절을 탐지하기 위해 문서 간의 겹침을 측정하는 대표적인 속성 계수법 방식들과 본 논문에서 제안하는 기법을 소개한다.[2)]

3.1 벡터 공간 모델(Vector Space Model)

정보 검색에서 질의와 문서 간의 유사도를 측정하는 다양한 방법이 있는데, 그 중 벡터 공간 모델은 질의와 문서를 각각 단어 벡터로 표현하여 두 벡터 사이의 유사도를 산출하여 검색된 문서들을 순위화하는 모델이다. n개의 유일한 단어가 출현하는 문서 집합에서 질의 Q와 문서 D_i를 n 차원의 가중치 된 단어 벡터$(W_1, \cdots, W_n)^T$로 표현할 때, 두 벡터 간의 유사도로는 식 (1)의 코사인 계수 값이 많이 사용된다.

$$sim(Q, D_i) = \frac{Q \cdot D_i}{\|Q\| \times \|D_i\|} = \frac{\sum_{j=1}^{n} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{n} w_{qj}^2} \times \sqrt{\sum_{j=1}^{n} w_{ij}^2}} \quad (1)$$

정보 검색의 벡터공간 모델을 표절 탐색 응용에 적용시키면, 질의와 문서 간의 유사도 측정은 질의 문서(주어진 문서)와 테스트 문서(비교 대상 문서)들 간의 유사도 측정으로 볼 수 있다. 예를 들어, 문서 집합의 유일한 단어들이 W = {a, b, c, d, e}이라고 할 때, 질의 문서 Q와 테스트 문서들 T_1, T_2, T_3, T_4는 다음과 같다고 가정한다 : Q = (a, b, c), T_1 = (a, b, c), T_2 = (a, b, c, d, e), T_3 = (a, b), T_4 = (a^k). 여기서 k는 테스트 문서 T_4에 발생하는 단어 a의 발생 빈도라 할 때, 질의 문서 Q와 가장 유사한 테스트 문서들은 T_1, T_2, T_3, T_4의 순서라고 볼 수 있다. 단어 벡터의 차원이 a, b, c, d, e의 순서라 가정하고 단어의 가중치가 단어 빈도로 부여 될 때, 질의 문서는 V(Q) = (1/3, 1/3, 1/3, 0, 0), 각 테스트 문서는 $V(T_1)$ = (1/3, 1/3, 1/3, 0, 0), $V(T_2)$ = (1/5, 1/5, 1/5, 1/5, 1/5), $V(T_3)$ = (1/2, 1/2, 0, 0, 0), $V(T_4)$ = (1, 0, 0, 0, 0)의 벡터로 표현된다.

두 벡터 간의 유사도를 간편하게 계산하기 위해 식 (1)의 분자만을, 즉, 벡터 내적만을 사용할 수도 있는데, 질의 문서와 테스트 문서 간의 유사도를 두 벡터의 내적 값으로 계산하면 Sim(Q, T_1) = 1/3×1/3+1/3×1/3+1/3×1/3 = 1/3이고, 유사한 방법으로 Sim(Q, T_2) = 1/5, Sim(Q, T_3) = 1/3, Sim(Q, T_4) = 1/3이 된다. 반면, 질의 문서와 테스

2) 제 3장의 기법들은 문서 뿐 아니라 문장 간의 표절을 탐지하기 위한 방법으로도 활용 될 수 있다.

트 문서 간의 유사도를 두 벡터의 코사인 계수 값으로 계산하면 Sim(Q, T_1) = (1/3)/($\sqrt{1/3} \times \sqrt{1/3}$) = 1이고, 유사한 방법으로 Sim(Q, T_2) = 0.77, Sim (Q, T_3) = 0.82이고, Sim(Q, T_4) = 0.58이 된다. 이 방식의 문제점은 Q와 T_2 간의 겹침이 Q와 T_3 또는 T_4와의 겹침보다 큼에도 불구하고 이를 제대로 반영하지 못한다는 점이다. 다른 문제점은 Q와 T_4와의 유사도를 계산 할 때, T_4에서 단어 a가 발생하는 빈도와 상관없이 항상 일정한 유사도 값을 갖는다는 점이다.

3.2 SCAM(Stanford Copy Analysis Mechanism)

단어들의 상대적 빈도를 고려하는 SCAM 기법은, 예를 들어, Q = (a, b, c)와 T_4 = (a^k)의 유사도를 비교 할 때, T_4에서 단어 a의 발생 빈도에 따라 다른 유사도 값을 갖도록 하였다. 예를 들어, T_4에서 a의 발생 빈도가 많을수록 Q와의 유사도는 낮아진다. 이를 위해, SCAM 기법은 비교하는 두 문서에서 발생 빈도가 서로 많이 다른 단어의 경우는, 해당 단어를 고려 대상에서 제외하기 위해 식 (2)의 근사 집합 C(Q, T)를 정의한다.

$$\in -\left(\frac{f_i(T)}{f_i(Q)} + \frac{f_i(Q)}{f_i(T)}\right) > 0 \qquad (2)$$

여기서, f_i(Q)와 f_i(T)는 단어 i가 질의 문서 Q와 테스트 문서 T에서 발생하는 빈도이며, 단어 i는 식 (2)를 만족하면 근사 집합 C(Q, T)에 포함된다. 또한 ∈는 2보다 큰 상수로, 큰 값은 근사 집합을 확장해 발생 빈도가 서로 많이 다른 단어도 고려 대상에 포함시키지만, 작은 값은 근사 집합을 축소해 발생 빈도가 서로 많이 다른 단어는 고려 대상에서 제외시킨다. 만약 두 비교 문서에서 단어 i의 발생 빈도가 서로 같다면 ∈의 값에 상관없이 단어 i는 근사 집합에 포함된다. 그러나 만약 f_i(Q) 또는 f_i(T)의 값이 0이면 근사 집합 조건 자체를 만족시키지 못하기 때문에 단어 i는 고려 대상에서 제외된다. 예를 들어, ∈가 3이고 테스트 문서 T_4에서 a의 발생 빈도가 1 또는 2일 때는 단어 a는 근사 집합에 포함되지만, 3보다 클 때는 단어 a는 근사 집합에서 제외된다.

SCAM 기법에서는 하나의 문서 D_1이 다른 문서 D_2의 부분 집합이거나 확대 집합이면 높은 유사도 값을 반환하도록 식 (3)의 부분집합 Subset (D_1, D_2)을 정의한다. 여기서 f_i(D_1)과 f_i(D_2)는 문서 D_1과 문서 D_2에서 단어 i의 발생 빈도이다.

$$Subset(D_1, D_2) = \frac{\sum_{w_i \in C(D_1, D_2)} f_i(D_1) \times f_i(D_2)}{\sum_{i=1}^{N} f_i(D_1) \times f_i(D_1)} \qquad (3)$$

SCAM 기법에서는 질의 문서 Q와 테스트 문서 T간의 유사도는 Subset(D_1, D_2)을 사용하여 식 (4)와 같이 정의된다. 단, 비교에 추가적인 정보를 주는 것이 아닌 1보다 큰 유사도 값은 1로 설정하기 때문에 유사도 값은 0~1 사이의 값을 갖는다.

$$Sim(Q, T) = \max[Subset(Q, T), Subset(T, Q)] \qquad (4)$$

따라서 SCAM 기법에서는, 예를 들어, ∈가 3이면, Sim(Q, T_1) = 1, Sim(Q, T_2) = 1, Sim(Q, T_3) = 1이 되지만 Sim(Q, T_4)의 경우는 k가 1일 때는 1 이고, k가 2일 때는 2/3이지만, k가 3보다 같거나 크면 0이 된다. 즉, 테스트 문서 T_4의 경우엔 단어 a의 빈도가 많을수록 유사도가 낮아진다. 하지만 이 방법에서는 한 문서가 다른 문서의 부분 집합일 경우 유사도가 1로 즉, Q와 T_1, T_2, T_3, T_4(k = 1)의 유사도가 동일하게 나오며, 또한 응용에 가장 적절한 ∈의 값을 설정해야 하는 문제점이 있다.

3.3 비트 시그니처(Bit Signature)

본 연구에서 제안하는 비트 시그니처 기법에서는 문서 집합의 유일한 단어들이 n개일 때 각 문서를 n개의 0 또는 1의 비트로 표현하되, 문서에 속한 단어는 1, 그 외의 단어는 0의 비트를 부여한다. 따라서 앞의 예와 같이 Q = (a, b, c), T_1 = (a, b, c), T_2 = (a, b, c, d, e), T_3 = (a, b), T_4 = (a^k) 일 때, 각 문서를 비트 시그니처로 표현하면 Q = (1, 1, 1, 0, 0), T_1 = (1, 1, 1, 0, 0), T_2 = (1, 1, 1, 1, 1), T_3 = (1, 1, 0, 0, 0), T_4 = (1, 0, 0, 0, 0)로 표현된다. 질의 문서 Q의 비트 시그니처를 Sig(Q), 테스트 문서 T의 비트 시그니처를 Sig(T)라고 할 때, 질의 문서 Q와 테스트 문서 T 간의 절대적 유사도는 식 (5)와 같다. 즉, 질의 문서 비트 시그니처와 테스트 문서 비트 시그니처 간에 bitwise AND 연산을 수행한 결과의 모든 비트들의 합을, 질의 문서의 유일한 단어들의 숫자로 나눠준다.

$$ASim(Q, T) = \frac{\sum(Sig(Q) bitwise\ AND\ Sig(T))}{\sum Sig(Q)} \quad (5)$$

따라서 앞의 예에서, 질의 문서와 테스트 문서들간 들의 절대적 유사도는 ASim(Q, T_1) = 3/3, ASim(Q, T_2) = 3/3, ASim(Q, T_3) = 2/3, ASim (Q, T_4) = 1/3이 된다. 하지만 이 방식에서 Q와 T_1과 T_2의 유사도를 비교할 때 Q와의 겹침의 크기가 같기 때문에, Q가 T_1과 더 유사함에도 불구하고 이를 제대로 구분하지 못하는 문제점이 발생한다. 어떤 두 쌍의 문서들 간의 유사도가 높은 가는, 절대적 겹침의 크기를 고려하는가 또는 상대적 겹침의 크기를 고려하는가에 따라 달라진다고 볼 수 있다. 상대적 겹침의 크기를 고려한다면, Q와 T_1간의 유사도는 3/3이지만 Q와 T_2 간의 유사도는 3/5로 Q와 T_1간의 유사도가 더 높다고 볼 수 있다. 따라서 비트 시그니처 기법에서는 유사도를 구할 때 절대적 겹침의 크기 뿐 아니라 상대적 겹침의 크기도 고려한다. 질의 문서 Q의 비트 시그니처를 Sig(Q), 테스트 문서 T의 비트 시그니처를 Sig(T)라고 할 때, 질의 문서 Q와 테스트 문서 T 간의 상대적 유사도는 식 (6)과 같다. 즉, 질의 문서 비트 시그니처와 테스트 문서 비트 시그니처 간에 bitwise AND 연산을 수행한 결과의 모든 비트들의 합을, 질의 문서 비트 시그니처와 테스트 문서 비트 시그니처 간에 bitwise OR 연산을 수행한 결과의 모든 비트들의 합으로 나눠준다.

$$RSim(Q, T) = \frac{\sum(Sig(Q) bitwise\ AND\ Sig(T))}{\sum(Sig(Q) bitwise\ OR\ Sig(T))} \quad (6)$$

앞의 예, Q = (a, b, c), T_1 = (a, b, c), T_2 = (a, b, c, d, e), T_3 = (a, b), T_4 = (a^k)에서, 질의 문서와 테스트 문서들 간에 절대적 유사도와 상대적 유사도를 함께 표시하면 Sim(Q, T_1) = (3/3, 3/3), Sim(Q, T_2) = (3/3, 3/5) Sim(Q, T_3) = (2/3, 2/3), Sim(Q, T_4) = (1/3, 1/3)이기 때문에, 절대적 유사도와 상대적 유사도를 함께 고려할 경우 Q와 T_1의 유사도가 Q와 T_2의 유사도 보다 높게 나옴을 알 수 있다. 따라서 비트 시그니처 기법에서의 유사도는 식 (7)과 같이 정의한다.

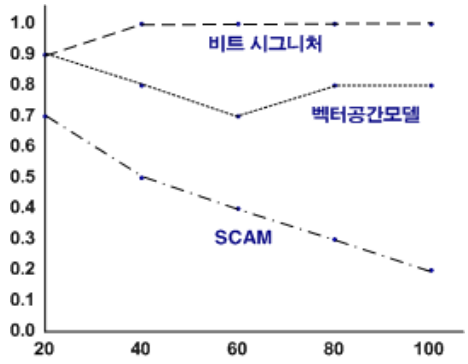
$$Sim(Q, T) = \alpha \times ASim(Q, T) + \beta \times RSim(Q, T) \quad (7)$$

여기서, α와 β는 0에서 1 사이의 가중치 값으로 α + β = 1이 된다. 앞의 예에 가중치 α와 β의 값을 0.5로 설정 할 때 비트 시그니처 기법의 유사도는 Sim(Q, T_1) = 15/15, Sim(Q, T_2) = 12/15, Sim(Q, T_3) = 10/15, Sim(Q, T_4) = 5/15로, 질의 문서 Q와 가장 유사한 테스트 문서들의 순서인 T_1, T_2, T_3, T_4로 제대로 순위화 함을 알 수 있다.

4. 실험 및 평가

본 연구에서는 문서를 전처리 한 후에 유사도 분석을 수행하였으며, 문서 전처리의 경우 카이스트의 한나눔 형태소 분석기를 사용하여 문서의 각 문장들을 형태소 별로 분리하였다.[3] 각 기법에 있어서 질의 문서를 기준으로 테스트 문서들을 표절 가능성이 높은 순서대로 순위화 하는지 조사하기 위해, 3명의 실험자가 구한 유사도 순위를 판단의 기준으로 사용하였다. 비트 시그니처 기법에서 α와 β는 각 0.5를 사용하였고, SCAM 기법의 경우 근사 집합을 결정하는 ∈는 실험적으로 가장 성능이 좋은 2.5를 사용하였다 [Shivakumar and Garcia-Molina, 1995].

첫 번째 실험은 질의 문서와 테스트 문서들 간에 내용의 유사도가 각 기법의 순위화에 어떤 영향을 미치는지 조사하였다. 이를 위해 질의 문서를 기준으로 유사한 내용이 0~20%, 0~40%, 0~60%, 0~80%, 0~100%인 각 10개의 테스트 문서를, 즉 질의 문서를 50개의 테스트 문서들과 비교하였다. 예를 들어, 질의 문서 "철수는 친구와 학교에서 집으로 간다"와 테스트 문서 "영희는 학교로부터 집까지 자동차로 간다"의 경우는, 질의 문서의 5개의 단어(또는 문장) 중 3개의 단어(또는 문장)가 테스트 문서에도 존재하므로, 테스트 문서는 질의 문서와 60%의 내용이 유사한 것으로 간주한다. <Figure 1>은 각 기법의 유사도 순위가 실험자가 구한 유사도 순위와 일치하는 비율을 보여준다. <Figure 1>에서 볼 수 있듯이 비트 시그니처 기법은 모든 경우에 실험자의 유사도 순위와 거의 일치함을 알 수 있다. 유사 내용이 0~20%일 때 오차가 생기는 이유는 비트 시그니처 기법의 경우 문서에 특정 단어가 포함될 때 그 단어의 빈도에 상관없이 한 비트로 표현되기 때문이다. 그러나 두 문서 간의 유사도 구하는데 있어서 이것으로 인한 문제는 실제적인 환경에서는 거의 발생하지 않는다. 예를 들어, 질의 문서 "철수와 영희는 학교에 간다"가 주어졌을 때 하나의 테스트 문서 "철수와 영희는 학교에 간다"와 다른 테스트 문서 "철수는 학교에 간다. 영희는 학교에 간다"와 같이, 두 테스트 문서의 단어 집합은 같지만 발생 빈도가 서로 다른 경우에만 비트 시그니처 기법은 두 테스트 문서의 유사도를 같은 것으로 간주하기 때문이다. 반면 SCAM 기법은 결과가 가장 안 좋은데, 이것은 SCAM 기법의 경우 한 문서가 다른 문서를 포함하는 경우 유사도를 일관되게 1로 결정해 순위화를 제대로 하지 못하기 때문이다. SCAM의 경우 두 문서 간에 유사한 내용의 비율이 많아질수록 두 문서 간의 포함 관계가 더 발생하기 때문에 성능이 더 나빠짐을 알 수 있다. 이 실험을 통하여 질의 문서가 주어졌을 때 비트 시그니처 기법은 테스트 문서들을 표절 가능성이 높은 순서대로 순위화 하는 것을 알 수 있다.

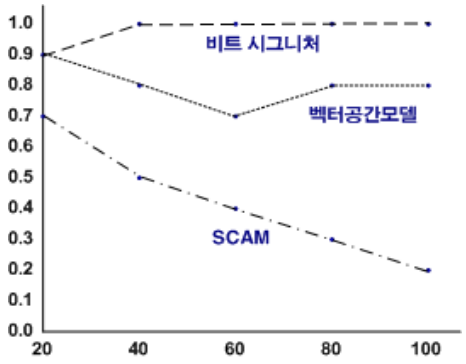


〈Figure 1〉 Similarity Ranking Performance of Each Method According to the Change of the Document's Similar Content

두 번째 실험은 테스트 문서의 구조 변경이 각 기법의 유사도 순위화에 어떤 영향을 미치는지 조사하였다. 예를 들어, 질의 문서 "철수는 친구와 학교에서 집으로 간다"를 기준으로, 두 테스트 문서 "철수는 친구와 학교에서 집으로 간다"와 "학교로부터 집까지 영희는 친구와 간다"의 경우, 두 문서의 내용은 80%가 유사하지만 문서의 구조는

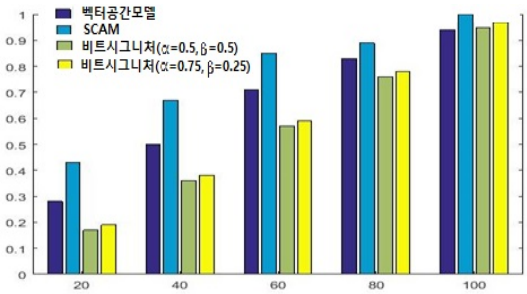
3) http://semanticweb.kaist.ac.kr/hannanum.

서로 다른 경우이다. 실험을 위하여 첫 번째 실험에서 사용한 테스트 문서들의 구조만을 바꾸어 사용하였으며, <Figure 2>는 각 기법의 유사도 순위가 실험자가 구한 유사도 순위와 일치하는 비율을 보여준다. <Figure 2>에서 볼 수 있듯이 각 기법의 유사도 순위화 결과는 <Figure 1>과 같음을 알 수 있다. 이것은 3가지 기법이 모두 문서의 구조에 상관없이 유사도 순위를 결정하기 때문이다. 즉, 비트 시그니처 기법은 문서의 단어 순서와 무관하게 단어들을 표현하기 때문에 속성 계수법 방식에 속하며, 따라서 문서의 구조 변경이 성능에 영향을 안 미치는 것을 알 수 있다.



〈Figure 2〉 Similarity Ranking Performance of Each Method According to the Change of the Document's Structure

세 번째 실험은 사용자가 추정하는 두 문서 간의 유사도와 각 기법의 유사도 값이 얼마나 근접한가를 조사하였다. 시스템이 산출하는 유사도 값이 사용자가 두 문서를 보고 추정하는 유사도와 가까울수록 시스템의 표절 탐지 성능이 더 우수하다고 볼 수 있다. 그러나 사실 사용자가 추정하는 두 문서 간의 유사도는 다소 주관적인 것이기 때문에, 질의 문서를 기준으로 테스트 문서의 절대적 겹침의 비율을 객관적인 척도로 사용하였다. 이를 위하여 질의 문서를 기준으로 20, 40, 60, 80, 100%의 절대적 겹침의 크기를 갖는 각 10개씩의 테스트 문서들을 사용하였다. 또한 테스트 문서는 질의 문서의 단어 집합 크기를 기준으로 ±50%의 단어 집합 크기를 갖는 문서들을 사용하였다. <Figure 3>은 각 절대적 겹침의 비율에 있어서, 왼쪽부터 벡터공간 모델, SCAM, 비트 시그니처(α = 0.5, β = 0.5), 비트 시그니처(α = 0.75, β = 0.25)의 유사도 값(0~1)을 보여준다. SCAM 기법은 한 문서가 다른 문서를 포함하면 일관되게 유사도 값 1을 반환하기 때문에, 사용자가 추정하는 유사도보다 매우 큰 값을 갖는 것을 알 수 있다. 반면, 벡터 공간 모델이나 비트 시그니처 기법은 사용자가 추정하는 유사도와 비슷한 유사도 값을 갖는 것을 알 수 있다. 특히 비트 시그니처 기법의 경우, 절대적 겹침의 크기를 조정하는 α를 크게 하면 사용자가 추정하는 두 문서 간의 유사도와 더욱 가까워짐을 알 수 있다. 즉, 비트 시그니처 기법의 α와 β의 비율을 즉, 절대적 겹침 크기나 상대적 겹침 크기의 반영 비율을 응용에 맞게 조정할 수 있는 장점이 있다.



〈Figure 3〉 Similarity of Each Method According to the Change of the Document's Absolute Overlap

5. 결 론

문서 저작권에 대한 중요성이 높아짐에 따라 표절 탐지에 대한 필요성이 증대되고 있다. 본 논문에서 제안하는 비트 시그니처 기법은 문서 간의 겹침을 측정하는 대표적인 속성 계수법 방식인 벡터 공간 모델이나 SCAM에 비해, 문서 표절 탐지를 위해 주어진 질의 문서와 가장 유사도가 높은 문서들을 정확히 순위화해 해주며, 사용

자가 두 문서를 보고 추정하는 유사도와 비슷한 유사도 값을 산출한다. 또한 비트 시그니처 기법은 문서의 단어순서와 무관하게 단어들을 표현하기 때문에 문서의 구조 변경이 성능에 영향을 미치지 않는 장점도 있다. 마지막으로 비트 시그니처 기법은 문서 간의 유사도 비교 뿐 아니라, 문장 안의 유일한 단어를 비트 시그니처 기법의 비트로 표현하여 문장 간의 유사도 비교에서도 사용할 수 있다. 따라서 표절 탐지 응용에서는 비트 시그니처 기법을 통하여 먼저 주어진 문서와 표절 가능성이 가장 높은 문서들을 찾은 후, 두 문서 간의 비교는 다시 비트 시그니처 기법을 사용하여 표절 가능성이 가장 높은 문장 쌍들을 찾아 나가는 방식으로 조사할 수 있다. 추후 과제로는 문서나 문장에 특정 단어가 여러 번 발생할 경우, 이를 비트 시그니처에서 적절한 방법으로 표현 해주는 방법에 대한 연구가 필요하다.

References

[1] Donaldson, J., Lancaster, A., and Sposato, P., “A plagiarism detection system”, In Proceedings of the 20th SIGCSE Technical Symposium on Computer Science Education, 1981.

[2] Gitchell, D. and Tran, N., “Sim : a utility for detecting similarity in computer programs”, In SIGCSE ’99 : The proceedings of the thirtieth SIGCSE technical symposium on Computer science education, 1999.

[3] Go, E. and Lee, J., “An Efficient Set-based POI Search Algorithm”, *Journal of KIISE : Computing Practices and Letters*, 2013.

[4] Go, E. and Lee, J, “Sentence Similarity Measurement Method Using a Set-based POI Data Search”, *KIISE Transactions on Computing Practices*, Vol. 20, No. 12, 2014, pp. 711-716.

[5] Ji, H., Jo. J., and Lim, H., “A Detection Method of Similar Sentences Considering Plagiarism Patterns of Korean Sentence”, *Journal of the Korean Association of Computer Education*, Vol. 13, No. 6, 2010, pp. 79-89.

[6] Jang, S., Seo, S., and Lee, K., “Clone Checker : A Program Similarity Checker”, Proceedings of the 28th KIISE Fall Conference, 2001.

[7] Kim, H. and Cho, H., “Improving Preprocessing step for Document retrieval system based on String Alignment”, Proceedings of the 35th KIISE Spring Conference, 2008.

[8] Ryu, C., Kim, H., and Cho, H., “Developing of Text Plagiarism Detection Model using Korean Corpus Data”, *KIISE Transactions on Computing Practices*, Vol. 14, No. 2, 2008, pp. 231-235.

[9] Salton, G., “The state of retrieval system evaluation”, *Information Processing and Management*, Vol. 28, No. 4, 1992, pp. 441-449.

[10] Schleimer, S., Wilkerson, D. S., and Aiken, A., “Winnowing : local algorithms for document fingerprinting”, In Proceedings of the 2003 ACM SIGMOD international Conference on Management of Data, 2003.

[11] Shivakumar, N. and Garcia-Molina, H., “SCAM : A copy detection mechanism for digital documents”, The 2nd International Conference on Theory and Practice of Digital Libraries, 1995.

[12] Wise, “YAP3 : Improved detection of similarities in computer program and other texts”, SIGCSEB : SIGCSE Bulletin, 1996.

저자소개

Woosaeng Kim

Woosaeng Kim is currently a professor of Computer Software department of Kwangwoon University. He received the bachelor's degree in the department of Computer Science from University of Texas at Austin. He received the MS and Ph. D. degree in the department of Computer Science from University of Minnesota. He had worked as a system engineer at Hyundai Electronics Co. His current research interests include database, multimedia, web application, etc.

Kyucheol Kang

Kyucheol Kang is currently a student of Computer Software department of Kwangwoon University.