# 대용량 문서 집합에서 유사 문서 탐색을 위한 전역 사전 생성 방법의 개선

Improvement of Creation Method of Global Dictionary for Finding Similar Texts among Massive Document Repository

박선영
Park Sun-Young
부산대학교 컴퓨터공학과
parksy@pusan.ac.kr

ABSTRACT

표절 검사 등을 위해 유사 문서를 탐색할 때, 탐색 대상 문서의 수가 많다면 탐색 시간이 매우 많이 걸린다. 이를 해결하기 위하여 전역 사전(Global DICtionary, GDIC)을 이용하여 전처리를 수행함으로써 성능을 개선하는 방법이 있다. 이 방법을 사용하면 신뢰도 80 ∼ 90 % 수준에서 탐색 시간이 기존 대비 10 ∼ 20 % 정도로 줄어든다. 이 방법을 사용한 탐색 시간 중 가장 큰 비중을 차지하는 것은 GDIC 생성 시간으로, 전체 탐색 시간의 50 ∼ 80% 정도를 소요한다. 본 보고서에서는 GDIC를 생성하는 방법을 개선하여 병목을 최소화하고 연산 시간을 감소시킴으로써 전체 연산 시간을 더욱 줄이는 방법을 제안한다. 우선 이미 생성되어 있는 STX, DIC에 대한 데이터베이스 저장 과정을 생략하여 Disk Access를 직접 수행하는 방법을 적용하여 불필요한 저장 시간을 없애고, GDIC 생성 과정에서 1차적으로 불용어 처리를 수행함으로써 GDIC 생성 시간을 대폭 줄일 수 있다. 또한 이 방법을 적용함으로 인해 GDIC의 크기도 대폭 줄어듦으로써 전처리 시간도 줄어드는 효과가 있다. 세종 계획에서 제공한 말뭉치 데이터와 국민일보에서 제공한 정부 보고서에 대한 실험 결과, STX, DIC 저장 과정을 생략함으로써 평균 4%, 생성 과정에서 불용어를 처리함으로써 평균 33% 정도 연산 시간이 감소하였으며, 두 방법을 모두 적용하였을 경우 35% 정도 연산 시간이 감소하였다. 이를 통해 전체 시스템의 탐색 시간을 25% 정도 줄일 수 있었다. 추후 동적인 문서 집합에도 유사한 방법을 적용할 수 있도록 개선할 계획이다.

KEYWORDS Plagiarism, Similar Document, DeVAC, Preprocessing, Global Dictionary

## 1 서 론

최근 정치계, 학계 유명인사의 표절 논란이 계속 불거지면서 2010년 3월경 저작권 위원회에서 내부에 표절 위원회를 구성하여 표절 문제를 전담하는 등 표절에 대한 관심이 높아지고 있다[1]. 표절이란 다른 사람의 저작물의 전부나 일부를 그대로 또는 그 형태나 내용에 다소 변경을 가하여 자신의 것으로 제공 또는 제시하는 행위를 의미한다[2]. 표절의 상당 부분을 차지하고 있는 문서 표절을 검사하기 위하여 문서 유사도 탐색 시스템에 대한 연구 개발이 진행되고 있다. 특히 최근에는 문서 집합이 다양화되고 대형화되면서 대용량 문서 집합에 대한 성능을 끌어올리기 위한 연구 개발이 진행되고 있다. 본 보고서에서는 전역 사전을 이용한 전처리 모델의 성능을 최적화하는 방법을 제안한다. 기존 방법에서 전역 사전에 포함시켰던 STX, DIC 구조에 대해 직접 디스크 접근으로 대체함으로써 생성

시간을 감소시키는 방법과, 전처리에 사용했던 불용어에 대한 처리를 사전 생성 시에 적용할 것이다. 이를 통해 생성 시간과 전역사전의 크기를 줄임으로써 사전 생성을 포함한 전처리 소요 시간을 감소시킬 수 있을 것이다.

## 2 관련 연구

유사문서 탐색 방법과 관련한 최근의 연구에는 원 저자권 개념에 기반한 뉴스 기사 표절 판정을 위한 프레임 워크에 관한 연구[3]와 BIBD 기반의 멀티미디어 핑거프린팅 코드와 공모코드들에 대한 공모자 추적[4]등이 있다. 또한 내용 기반 유사 문서 탐색 시스템 DeVAC[5]은 문서의 길이에 관계없이 강력한 탐색 성능을 수행하는 시스템으로, Attribute Counting[?] 중 하나인 Fingerprint[6]와 Structured Metric[7]을 사용한다. 또한 대용량 문서 집합에서의 성능을 끌어올리기 위하여 전역 사전(Global DICtionary, GDIC)에 기반한 전처리 방법[8]을 제안하였다. 이 전처리 방법에서는 불용어를 걸러내는 비율을 결정하는 $T_{ratio}$, 유사 문서 쌍 후보를 찾기 위해 필요한 최소 공통 키 등장 횟수 $N_{match}$, 두 문서 간의 공통 키 등장 횟수 합계를 나타내는 $S_{match}$, 후보로 등록하기 위한 공통 키 등장 비율의 기준을 나타내는 $C_{ratio}$ 등의 환경 변수를 사용하여 전처리를 수행하며, 전처리를 수행함으로써 5천건 이상의 대용량 문서 집합에 대해 검사해야 할 문서 쌍의 개수를 기존 방법의 5 $\sim$ 10% 수준으로 감소시킬 수 있다는 것을 확인했고, 또 최근에는 이 모델을 DeVAC에 실제로 적용하여 전체 시스템 연산 시간을 기존 방법의 10% $\sim$ 15% 수준으로 감소시킬 수 있다는 것을 확인하였다.

## 3 전처리 시스템의 병목 분석 및 개선점 연구

### 3.1 전처리 시스템의 연산 시간 및 병목 분석

위에서 설명한 GDIC를 이용한 전처리 방법은 그 성능 향상의 폭이 크고, Similarity도 80 $\sim$ 90 % 정도로 높은 편이다. 하지만 여기에서 연산 시간을 더욱 줄일 수 있는 방법을 찾을 수 있다. 우선 이 방법의 연산 시간에서의 병목을 분석할 필요가 있다. GDIC를 이용한 전처리 방법을 사용한 전처리에서 주목할 점은, 전체 유사 문서 탐색 시간에서 GDIC 생성 시간이 가장 큰 비중을 차지한다는 것이다. 표 1 은 전처리 모델을 적용했을 때 각 과정에서 소요되는 시간의 비중을 나타낸 것이다.

표에서 보면 알 수 있듯이 GDIC 생성 시간은 대부분의 실험 조건에서 전체 연산 시간의 50 $\sim$ 80%정도의 비중을 차지하고 있음을 알 수 있다. 이 수치는 매우 큰 의미가 있다. 비록 GDIC가 재사용이 가능하다 하더라도 그것은 동일 문서 집합 내에서 또다른 검사를 수행할 때 가능한 것이고 새로운 문서 집합에 대한 연산을 수행한다면 각 문서집합마다 GDIC를 생성해야 하므로, GDIC 생성 시간을 줄인다면 전체 탐색 시간도 감소할 것이다.

### 3.2 GDIC의 구조

GDIC 생성 시간을 최소화하기 위한 방법을 찾기 전에 기존의 GDIC 자료 구조의 상세한 정보와 생성 과정에 대해 살펴볼 필요가 있다. GDIC는 STX, DIC, GDIC summeries(GDS), GDIC details(GDD)

| 실험 조건 | GDIC 생성 시간 | 전처리 시간 | 검사 시간 | 전체 탐색 시간 |
|---|---|---|---|---|
| 조건1 | 12,473(63.1) | 2,189(11.1) | 5,103(25.8) | 14,662.02(100) |
| 조건2 | 12,473(85.3) | 733 (5.0) | 1,414 (9.6) | 13,207.05(100) |
| 조건3 | 12,473(82.7) | 781 (5.1) | 1,825(12.1) | 13,254.18(100) |
| 조건4 | 12,473(71.4) | 938 (5.4) | 4,052(23.2) | 13,410.88(100) |

표 1. 약 6200건의 보고서에 대해 전처리 방법을 총 4가지 실험 조건에서 측정한 결과. 단위는 초. 괄호 안은 전체 탐색 시간에 대한 비율(%). GDIC 생성 시간은 모든 조건에 대해 동일하다. 전체 탐색 시간에서의 비중을 살펴 보면 모든 실험 조건에서 GDIC 생성 시간이 가장 큰 비중을 차지하고 있음을 알 수 있다.

등의 세부 자료 구조로 나뉜다. 이 중 STX와 DIC는 검사 파일 생성 시점에서 이미 생성되어 있는 것으로, DeVAC에서 정밀 검사를 수행할 때 사용되는 자료 구조이다. GDIC summeries는 각 Key에 대한 개략적인 정보를 표시한 것으로, 불용어 처리에 사용한다. GDIC detials는 각 Key에 대한 세부적인 index 정보를 모두 기록한 것으로, 전처리에 사용한다. 위의 두 자료구조는 모두 DIC로부터 생성되며, STX와 DIC를 모두 GDIC 내부에 저장한 이유는 원본 STX와 DIC 파일이 손상되더라도 정상적인 탐색을 수행할 수 있도록 하기 위해서이다.

### 3.3 전역 사전 생성 개선 방법

위에서 분석한 GDIC의 구조를 바탕으로 제안하는 전역 사전 생성 개선 방법은 다음의 두 가지이다.

1. STX와 DIC 데이터는 기존에 파일로도 존재하지만 이를 GDIC 내부에 중복으로 저장한다. 실질적으로 파일이 손상되는 경우는 극히 드물기 때문에, 파일 경로에 대한 저장으로 대체할 수 있다. 이렇게 할 경우 대용량 문서 집합에 문서가 5천건, 한 파일의 크기가 평균 250KB 정도라면 약 1.3GB 크기에 해당하는 Disk Access를 생략함으로써 생성 시간을 단축할 수 있다.

2. 현재 GDIC details의 경우 모든 범위의 Key index를 저장하고 있다. GDIC summeries는 기존 방법대로 저장하되, 이를 이용해 일반적인 전처리 범위보다 조금 넓은 범위를 적용하여 GDIC details 생성 전에 Key에 대한 전처리를 수행할 수 있다. 이전 연구를 통해 밝혀진 적절한 $T_{ratio}$ 값의 범위가 0.003 ~ 0.005 인 것을 감안해 약간 더 넓은 범위를 적용, 0.010 정도의 값으로 적용하여 GDIC 생성 수준에서 전처리를 수행하는 것이다. 이렇게 할 경우 GDIC details의 크기도 작아져 Key 탐색 시간이 줄어들어 전처리 속도 역시 빨라지게 된다.

위의 두 방법을 각각 적용하고, 또 두 방법을 동시에 적용하면 성능이 향상될 것으로 기대된다. 성능 향상 정도는 실험을 통해 측정할 것이다.

## 4 실 험

### 4.1 실험 데이터 및 환경

실험에 사용된 데이터는 크게 두 가지이다. 첫 번째는 1999 ∼ 2009 년의 정부 정책 보고서 6,808 건 중 10 ∼ 120,000 개의 어절로 이루어진 6,263 건의 아래아 한글 문서를 사용하였다. 총 용량은 1.52GB 이며 문서 하나의 평균 크기는 250KB 이다. 한글 문서이므로 순수 텍스트 용량은 이보다 작다. 두 번째 데이터는 세종계획[9] 에서 제공한 실험용 말뭉치 데이터를 가공한 것으로, 총 20,000 건이다. 순수 텍스트 총 용량은 277MB 이다. 실험은 Intel Xeon 서버 머신 (DeVAC 서버) 으로 진행하였으며, 머신 상세 사양은 표 2 과 같다.

| 구분 | 성능 |
|---|---|
| CPU | Intel Xeon 2000MHz |
| RAM | 4096 MB |
| GPU | Geforce GT 6600 |
| HDD | 300 GB × 2 |

표 2. (실험에 사용한 서버 머신의 사양 표)

### 4.2 실험 방법

실험은 위의 두 가지 실험 데이터에 대하여 각각 다음의 4 가지 조건에 대해 수행한다.

A) STX, DIC, GDIC summary(GDS), GDIC details(GDD) 모두 저장 (기존의 방법)

B) STX, DIC 에 대해 기존 file path 만 저장하는 방법 적용 (GDS, GDD 는 그대로 저장)

C) GDD 에 대해 불용어 처리 수행 (STX, DIC, GDS 는 그대로 저장)

D) 방법 B 와 방법 C 를 동시에 적용

기존 방법에서 GDIC 전체 생성 과정은 다음과 같다. 테이블 생성 후 STX 를 DB 에 복사한 후, DIC 를 읽어들이는 과정에서 GDS 를 구축하기 위한 Counting 과 GDD 생성을 수행한다. 수행이 끝나면 DIC, GDD, GDS 를 모두 저장한다. b 방법은 STX 에 대해서는 경로 정보만을 저장하고, DIC 는 읽어들인 후 GDD 와 GDS 생성에만 사용하고 DB 에 쓰지는 않는다. c 방법은 STX, DIC 저장은 기존 방법 그대로 수행하지만, GDS 구축에서 Counting 을 수행한 후, Counting 결과에 따라 일정 비율 이상 사용된 Key 를 불용어 처리하여 저장하지 않는다는 점이 다르다. 여기서 비율은 실험을 통해 찾아냈던 $T_{ratio}$ 의 적절 범위인 0.003 ∼ 0.005 를 고려하여 0.010 로 정한다. d 방법은 b, c 에 사용된 방법들을 동시에 적용한다. A, B, C, D 의 방법을 정부 보고서 데이터와 말뭉치 데이터에 대해 각각 적용하여 시간을 측정 비교한다. 연산 시간의 측정이 끝나면 생성된 GDIC 의 각 세부 항목의 크기와 전체 크기를 조사하여 방법 별로 비교한다.

### 4.3 실험 결과

정부 보고서와 말뭉치 데이터에 대해 4가지 방법을 적용하여 연산 시간을 측정한 결과는 표 3, 그림 1과 같다. 우선 본 보고서에서 개선을 목표로 했던 GDIC 생성 시간의 경우 방법 B를 사용하였을 때 방법 A(기존 방법)에 비해 3 ~ 5% 정도 시간이 줄어든 것을 확인할 수 있다. 방법 C를 적용할 경우에는 30 ~ 35 % 정도로 GDIC 생성 시간이 크게 개선되며, 두 방법 모두를 적용하면 35 ~ 38 % 정도로 GDIC 생성 시간이 더욱 줄어드는 것을 확인할 수 있다. 더불어 방법 C에서 전처리 시간도 조금 감소한 것을 확인할 수 있는데, 이는 GDD의 크기가 대폭 감소하면서 DB의 처리 시간이 짧아진 것이 원인으로 판단된다. 즉 DB의 크기가 탐색 시간에 미세한 영향을 미친 것이다. 또한 전처리 과정은 기존의 방법과 동일하기 때문에 전처리 후 문서쌍 역시 동일할 것이고, 이에 따라 검사 시간은 거의 일정하게 측정된 것을 확인할 수 있다. 최종적으로는 GDIC 생성 시간 감소로 인해 기존 방법보다 전체 탐색 성능이 25% 가량 향상된 것을 확인할 수 있었다. 이는 그림 1을 보면 확인할 수 있다.

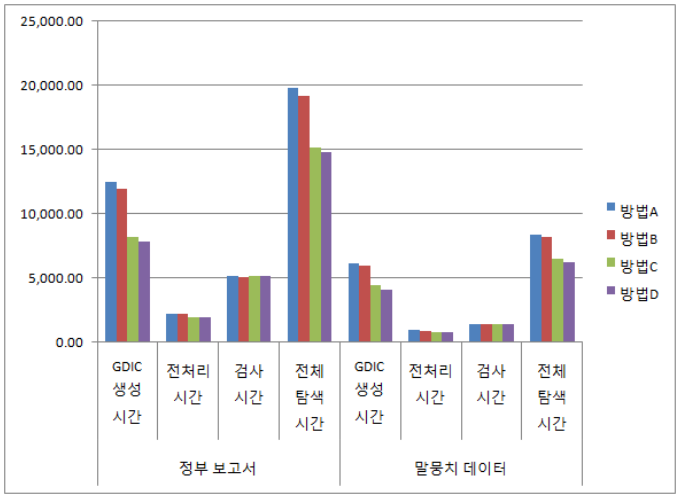


그림 1. 표3의 결과를 나타낸 그래프. 전처리 시간과 검사 시간은 모든 방법에 대해 거의 동일하지만, GDIC 생성 시간이 줄어드는 것을 확인할 수 있다. 또한 그 차이가 전체 탐색 시간에 반영되어 있음을 알 수 있다.

처리시간의 측정이 끝난 후 완성된 GDIC의 크기를 비교해 보았다. 그 결과는 표 4와 같다. 기존 방법에서 가장 큰 비중을 차지하는 데이터는 STX, DIC였는데 방법 B로 이를 제거하자 GDIC 전체 크기가 대폭 감소한 것을 확인할 수 있다. 방법 C는 GDD 크기 감소에 효과가 있으며, GDS는 변하지 않는다. 전체 합을 놓고 보았을 때는 방법 C보다는 방법 B가 더욱 큰 효과가 있으며, 연산 속도 실험때와 마찬가지로 두 방법을 동시에 적용했을 때 크기 감소 효과가 가장 큰 것으로 나타났다.

## 5 결론

본 논문에서는 DeVAC 전처리 모듈의 성능을 개선하기 위하여 전처리 과정 중 가장 큰 비중을 차지하는 GDIC 생성 시간을 감소시키기 위한 방법들을 제안하였고, 실험을 통하여 GDIC 생성 시간 감소

| | | 방법 A (기존) | 방법 B (개선 1) | 방법 C (개선 2) | 방법 D (동시적용) |
|---|---|---|---|---|---|
| 정부 보고서 | GDIC 생성 시간 | 12,478.31 (100) | 11,907.17 (95.42) | 8171.34 (65.48) | 7812.97 (62.61) |
| | 전처리 시간 | 2,189.51 (100) | 2,161.82 (98.74) | 1,871.15 (85.46) | 1881.49 (85.93) |
| | 검사 시간 | 5,103.36 (100) | 5,090.06 (99.54) | 5093.02 (100.19) | 5,091.32 (99.76) |
| | 전체 탐색 시간 | 19,771.18 (100) | 19,149.05 (96.85) | 15,386.18 (76.65) | 14,785.78 (74.78) |
| 말뭉치 데이터 | GDIC 생성 시간 | 6,120.24 (100) | 5,970.93 (97.56) | 4376.8 (71.51) | 4097.09 (66.94) |
| | 전처리 시간 | 899.58 (100) | 871.66 (96.89) | 791.24 (87.96) | 788.31 (87.63) |
| | 검사 시간 | 1,352.62 (100) | 1361.31 (100.64) | 1344.42 (99.39) | 1355.89 (100.24) |
| | 전체 탐색 시간 | 8,372.44 (100) | 8,203.90 (97.99) | 6,512.46 (77.78) | 6,241.29 (74.55) |

표 3. 정부 보고서와 말뭉치 데이터에 대해 총 4가지 방법으로 전역 사전 생성 및 탐색을 수행한 결과. 괄호 안은 기존 방법을 100으로 했을 때의 비율. GDIC 생성 시간의 경우 방법 B를 사용하였을 때 방법 A(기존 방법)에 비해 3 ~ 5%, 방법 C를 적용할 경우 30 ~ 35 %, 두 방법 동시 적용 시 35 ~ 38 % 정도로 GDIC 생성 시간이 개선되는 것을 확인할 수 있다. 방법 C에서 전처리 시간이 약간 줄어든 것은 GDD의 크기 감소로 인해 DB 처리 시간이 줄어든 데 따른 것이다. 전처리 과정은 동일하므로 검사 시간은 일정하며, 전체 탐색 시간은 평균적으로 25%가량 향상된 것을 확인할 수 있다.

및 전체 시스템 향상 정도를 측정하였다. 실험 결과는 다음과 같다.

1. GDIC 생성 시간에 대해서 STX, DIC를 저장하지 않았을 때 평균 3 ~ 5% 의 성능 향상이 있었다.
2. GDIC 생성에 불용어 처리를 적용하였을 때 33% 의 성능 향상이 있었다.
3. GDIC 생성에 위의 두 방법을 동시에 적용하였을 때 35% 정도의 성능 향상이 있었다.
4. 이 방법을 실제 시스템에 적용하여 측정하였을 경우, 1회 검사에 25% 정도 성능이 향상되는 것을 확인할 수 있었다.

| | | 방법 A (기존) | 방법 B (개선 1) | 방법 C (개선 2) | 방법 D (동시적용) |
|---|---|---|---|---|---|
| 정부 보고서 | STX, DIC | 719.60 | 0.195 | 719.6 | 0.195 |
| | GDD | 191.20 | 191.20 | 80.06 | 80.06 |
| | GDS | 28.9 | 28.9 | 28.9 | 28.9 |
| | GDIC | 939.70 | 220.30 | 828.56 | 109.12 |
| 말뭉치 데이터 | STX, DIC | 468.20 | 0.612 | 468.20 | 0.612 |
| | GDD | 309.40 | 309.40 | 157.32 | 157.32 |
| | GDS | 60.60 | 60.60 | 60.60 | 60.60 |
| | GDIC | 838.20 | 370.61 | 686.12 | 218.53 |

표 4. 정부 보고서와 말뭉치 데이터에 대해 총 4가지 방법으로 전역 사전 생성 및 탐색을 수행 후 GDIC의 전체 크기. 단위는 MB(메가바이트). 방법 B로 STX, DIC의 크기를, 방법 C로 GDD의 크기를 대폭 줄일 수 있으며 두 방법을 모두 적용하면 GDIC 전체의 크기를 절반 이하로 대폭 줄일 수 있다.

이 방법은 불용어를 처리하는 과정을 거친 후 GDIC를 생성하기 때문에 정적인 문서 집합에만 적용이 가능하며, 문서가 지속적으로 삽입/삭제되는 경우에는 적용하기 힘든 방법이다. 추후 이러한 단점을 개선하여 문서 집합의 삽입이 지속적으로 이루어지는 경우에도 이 방법을 적용할 수 있도록 시스템을 개선할 계획이다.

## 참고 문헌


1. "특허청," http://www.kipo.go.kr/, 2010.
2. "저작권 위원회," http://www.copyright.or.kr/, 2010.
3. 김정민, 정현숙, 이종영, 강남준, "뉴스 기사 표절 판정을 위한 시스템 프레임워크," in *Proc. of the KIIT*, 2009, pp. 228–233.
4. 이강현, "Bibd 기반의 멀티미디어 핑거프린팅 코드의 공모코드들에 대한 공모자 추적," in *Proc. of the IEEK*, 2009, pp. 79–86.
5. 류창건, 김형준, 조환규, "한글 말뭉치를 이용한 한글 표절 탐색 모델 개발," in *Proc. of the KIISE*, 2008, vol. 14, pp. 231–235.
6. S. Schleimer, D. S. Wikerson, and A. Aiken, "Winnowing : local algorithms for document fingerprinting," in *Proc. of the ACM SIGMOD international conference on Management of data*. 2003, pp. 76–85, ACM.
7. A. Apostolico, "The myriad virtues of subword trees," *Combinatorial Algorithms on Words*, vol. 37, no. 3, pp. 85–96, 1985.
8. 박선영, 김지훈, 김선영, 김형준, 조환규, "대용량 문서 집합에서 유사 문서 탐색을 위한 효과적인 전처리 시스템의 설계," in *Proc. of the KIISE*, 2009, vol. 36, pp. 76–77.
9. "21세기 세종 계획," http://www.sejong.or.kr/, 2010.
10. J. L. Donaldson, A. Lancaster, and P.H. Sposato, "A plagiarism detection system," in *Proc. of the Twelfth SIGCSE Technical Symposium on Computer Science Education*, 1981, pp. 21–25.