한국컴퓨터정보학회 하계학술대회 논문집 제18권 제2호 (2010. 7)

문서구조 정보 기반의 유사도 측정

신미해°, 고방원**, 김영철***, 정진영****
°공주대학교 컴퓨터과학과 석사과정
**숭실대학교 IT 대학 컴퓨터학과 박사과정
***유한대학 e-비즈니스과 교수
****대전보건대학 바이오정보과 교수
e-mail: tlsalgo@nate.com, withfox@ss.ssu.ac.kr, kim0725@yuhan.ac.kr, jyjung@hit.ac.kr

A Similarity Evaluation using Structural Information of Documents

Mi-Hae Shin°, Bang-Won Ko**, Young-Chul Kim***
°Dept. of Computer Science, KongJu University
**Dept. of Computer, SoongSil University
***Dept. of e-Business, Yuhan University
****Dept. of Bio Information

◉ 본논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임("KRF-2007-331-D00435")

● 요 약 ●

인터넷의 발달로 인한 수많은 정보의 공유는 지식 정보사회의 발전을 가져왔다. 이러한 정보사회의 발전과 동시에 표절과 같은 새로운 지식 범죄도 급증하고 있다. 표절은 연구의 정직성과 창의성을 떨어뜨리고 학문의 발전을 저해하는 요소이다. 이러한 표절을 근절하기 위해서 그동안 많은 방법들과 시스템들이 제시되었다. 이중 자연어로 구성된 구조가 없는 일반 문서의 표절을 검사하는 방법은 지문법을 이용하였다. 지문법과 같이 통계적인 방법을 이용한 유사도 검사 방법은 문서 대 문서 전체를 비교하기 때문에 부분적 유사성, 즉 문장이나 문단 단위의 비교를 할 수 없는 단점이 있다. 본 논문에서 제시하는 시스템은 자연어로 이루어진 일반문서 중 특별한 문서의 구조 정보를 가질 수 있는 일반 텍스트 문서를 대상으로 유사도를 측정 하였다. 즉 텍스트 문서 구조를 AST 형태의 자료구조로 표시하고 이를 이용하여 사용자가 원하는 부분 또는 전체 유사도 측정 방법을 제시한다.

키워드: 표절(Plagiarism), 지문법(fingerprint), 유사성(Similarity)

I. 서론

일반적으로 문서 표절 검사를 할 때, 구조적인 특징보다는 통계적인 특징을 추출하여 유사도를 검사하며 이러한 기법을 지문법(fingerprint)이라 한다. 지문법을 쓰는 이유는 보통의 일반 텍스트 문서들은 인공 언어와는 달리 구조를 지니고 있지 않은 문서이기 때문이다[1]. 따라서 지문법은 문서 내에 등장하는 단어들의 빈도수와 가중치를 이용하여 유사도를 구하는 방법이다. 하지만 통계적인 수치만을 이용했을 경우 두 문서의 전체적인 유사도만을 알 수 있기 때문에 실제로 문서의 어느 부분이 유사한지를 알기 어렵다는 단점이 있다. 또한 서로 다른 문서임에도 불구하고 우연히 동일한 단어가 많이 등장했을 때 유사성이 높게 나오는 단점을 가지고 있다[2]. 본 논문에서 제시하는 방법은 일반 텍스트 문서 중 일정 형식을 가지고 있는 논문을 구조화 시켜 XML 문서로 표현한 후 트리를 비교하는 방법을 제시한다. 이 방법은 기존의 통계적 분석 방법과는 달리 문서를 구조화 시켜서 분석할 수 있다는 장점이 있으며 구조 분석을 통하여 어느 부분이 실제로 일치하는지 알 수 있고 우연히 동일한 단어가 등장하더라도 구조가 다르기 때문에 유사하지 않은 것으로 판단할 수 있다.

II. 관련 연구

지문법에는 거리를 이용한 방법과 상관계수를 이용한 방법 두 가지가 있다.

1. 거리를 이용한 방법

객체간의 유사성의 정도를 정량적으로 나타내기 위해서 척도가 필요하다. 가장 보편적으로 많이 사용되는 것이 거리(distance)인데 거리와 같이 클수록 유사성이 적어지는 척도는 비유사성 척도(dissimilarity measure)라 한다. 거리를 이용한 유사도 측정 방법에 유클리안 거리(Euclidean distance), 민코프스키 거리(Minkowski distance), 맨하탄 거리(Manhattan distance)[3] 등이 있다. 이중

가장 많이 사용되는 유클리드 거리는 L2 Distance라고도 불리며 이는 민코프스키 거리(Lm Distance)에서 m이 2인 특별한 경우이다. 유클리디안 거리는 두 점의 좌표를 n 차원으로 생각하여 거리를 계산하는 방법이다. 두 점 P와 Q가 각각 $P = (p_1, p_2, p_3, \cdots, p_n)$와 $Q = (q_1, q_2, q_3, \cdots, q_n)$의 좌표를 갖을 때 두 점사이의 거리를 계산한다.

2. 상관 계수를 이용한 방법

상관계수를 이용하는 방법은 문서와 질의어를 벡터로 표현하고 여기에 가중치를 부여한 다음 유사도를 측정하는 방법이다. 상관계수에는 코사인 계수, 자카드 계수, 피어슨 상관 계수 등이 있다[4]. 이와 같은 상관 계수를 이용하여 유사도를 측정하는 방법은 크게 세 단계로 구분할 수 있다.

우선 문서에서 추출된 단어(term)들의 색인 작업을 하는 것이다. 색인 작업을 하기 위해서는 문서에 표현된 중요하지 않은 단어들(기능어)을 자동문서 색인을 통해서 벡터에서 제거해야 한다. 그러면 그 문서는 관련 내용어로만 표현된다[5]. 이러한 색인은 단어의 빈도를 기본으로 하고, 문서 내에서 높고 낮은 빈도를 가지는 단어는 기능어로 취급한다[5, 6].

III. 본론

본 논문에서 제안 하는 유사도 평가 시스템의 전체 구조는 그림 1과 같이 문서 변환, 문서 구조 분석(parsing), 유사도 측정 과정으로 크게 세 부분으로 나눌 수 있다.

우선 문서 변환부분은 doc, hwp, pdf 등과 같은 형태의 문서를 구조적으로 표현하기 위해서 XML 문서로 변환해주는 역할을 한다. 이렇게 변환된 XML문서를 구조적으로 분석하기 위해서 파싱 과정을 거친다. 그리고 마지막으로 본 시스템에서 제안하는 유사도 알고리즘을 이용하여 유사도를 측정한다.

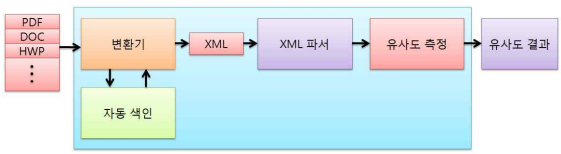

그림 1. 유사도 평가 시스템의 구조
Fig.1. Architecture of Similarity Evaluation System

1. 문서 변환기

일반 텍스트로 이루어진 문서들은 구조를 가지고 있지 않은데 이 중에는 특정 형식을 가지는 문서가 있다. 예를 들면 논문과 같은 문서는 일정한 양식을 가지고 있는 문서로써, 학회, 학술, 학위 등 그 종류에 따라 다양한 형태의 구조를 띄고 있다. 또한 각각의 구조를 나타내는 메뉴 디스크립션(Menu Description)을 제공하고 있으며, 논문 제출자는 메뉴 디스크립션의 형태에 맞게 작성하여 제출하도록 되어있다. 문서 변환기는 이러한 구조적 정보를 XML 스키마 형태로 표현하고 이를 이용하여 일반 문서를 XML 문서로 변환하는 기능을 한다.

2. 자동 색인

자동 색인은 "색인어로 선택될 가능성이 있는 모든 용어를 추출하여 후보 색인어를 생성하는 과정" 과 "후보 색인어로부터 불용어를 제거하고 색인어를 선별하는 과정" 으로 이루어진다[7]. 즉 형태소 분석을 끝낸 후 모든 용어를 색인하는 것이 아닌 실제로 의미가 있는 용어만을 추출하는 과정으로, 본 논문에서는 부분적인 형태소 분석 기법을 사용하여 자동 색인을 구성하였다. 부분적인 형태소 분석 기법은 문서내의 명사만을 추출하는 것으로 명사 이외의 조사, 형용사, 동사를 불용어로 간주하고 색인 후보에서 제외한다. 명사를 색인어 집합으로 선택한 이유는 대부분의 의미는 명사에 의해 전달되기 때문이다. 이렇게 문서내의 불용어들을 제거하면 총 단어 수의 40~50% 제거 되어 색인어의 수를 줄일 수 있다[5]. 색인 테이블의 구조는 명사를 키로 갖고, 색인을 값으로 갖는 해시테이블로 구성되어있다. 인덱스 테이블을 구성하는 과정은 그림 2와 같다.

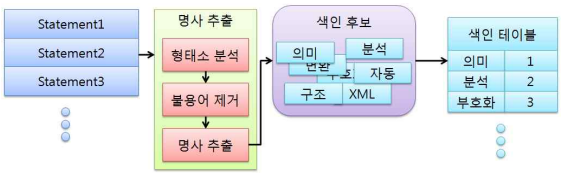

그림 2. 인덱스 테이블
Fig. 2. Index Table

그림 2의 과정을 보면 문장 단위로 형태소 분석기에 입력하여 불용어들을 제거하고 명사들을 추출한다. 이렇게 생성된 명사들을 색인 테이블에 입력함으로써 색인테이블을 구성한다. 예를 들어 "의미 분석과 부호화된 구조 분석을 이용한 XML 자동 변환" 문장을 형태소 분석을 거쳐 명사를 추출하면 "의미", "분석", "부호화", "구조" 등의 명사들이 추출된다. 이렇게 추출된 명사들을 색인테이블에 (색인, 인덱스)의 쌍으로 삽입하게 된다.

3. XML 파서

기존의 잘 알려진 XML 파서에는 DOM과 SAX가 있다. 이중 DOM은 메모리를 차지하는 크기나 수행속도가 느린 단점이 있다. 따라서 본 논문에서는 SAX를 이용하여 XML 문서를 파싱한다. 하지만 SAX는 DOM과 같이 트리 형태의 자료구조를 만들어 주지는 않는다. 따라서 SAX API와 stack을 이용하여 AST(abstract syntax tree)를 구성하였다. AST는 파스 트리와는 달리 모든 심볼에 대하여 노드를 만들지 않고 최소한의 노드를 만드는 트리를 말한다. 이렇게 만들어진 트리는 최소한의 노드만을 가지고 있기 때문에 공간과 시간을 절약할 수 있는 장점이 있다. 본 시스템에서 사용된 AST 노드의 자료 구조는 표 1과 같다.

표 1. AST 노드의 자료구조
Table 1. Data Structure of AST Node

```
public class NounNode{
   private Vector<Node> m_childNode = new Vector<Node>();
   private String m_nodeName;
   private String m_PCData;
   private boolean m_check;
   private int m_index;
}
```

표 1의 노드 자료구조는 추상 클래스(abstract class)로서 모든 노드가 상속받으며, m_childNode 는 Vector 자료형으로 n개의 자식을 가질 수 있도록 구성하였다. n개의 자식을 가질 수 있도록 한 이유는 한 논문에 등장하는 문단(paragraph), 문장(statement), 명사(noun)가 얼마나 등장하는지 알 수 없기 때문이다. 또한 m_nodeName은 태그명으로써 현재 어떤 노드를 검사하고 있는지를 알 수 있고 m_check는 현재 노드가 검사되었는지 안 되었는지를 검사할 때 사용된다. m_PCData는 실제 명사데이터의 정보이고 이를 이용하여 인덱스 테이블에서 검색하여 index를 찾아 m_index 에저장한다.

4. 유사도 측정

AST 생성과정이 끝나면 비교하려는 두 문서의 AST를 이용하여 유사도를 측정한다. 유사도를 측정하기 위해서는 AST 자료구조를 선형의 자료 구조로 변환해야 한다. 그림 3은 AST에서 선형 자료구조로 변환하는 방법을 보여준다.

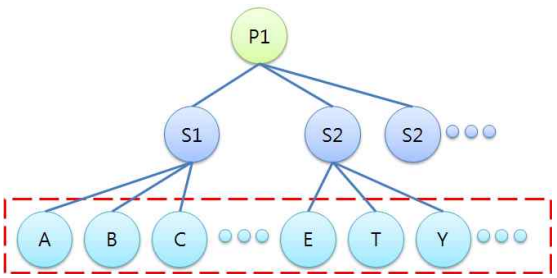

그림 3. 명사의 선형 자료구조 추출
Fig. 3. Linear Data Structure Extraction of Noun

그림 3에서 P1은 문단을 의미하며 S1은 문장을 의미한다. S1 단위로 명사들을 추출하여 선형의 자료구조를 만든다. 이렇게 생성된 선형자료구조를 앞으로 시퀀스(sequence)라 정의한다. 점선으로 표현된 부분은 이러한 문장단위의 시퀀스를 이어놓은 것으로 한 문단의 전체 시퀀스를 나타낸다. 유사도 평가 알고리즘은 일치하는 서브 시퀀스 찾기(search), 마킹(marking), 유사도 측정(measure)의 세 단계로 구성된다. 첫째, 일치하는 시퀀스 찾기에서는 두 시퀀스 A, B에서 일치하는 서브 시퀀스를 찾는 것이다. 이 때 일치하는 여러 서브 시퀀스가 있다면 가장 긴 서브 시퀀스를 찾는 것이 목적이다. 예를 들면 시퀀스 A, B가 그림 4와 같다면 matchSize 값은 5(11, 55, 2, 1, 85)와 3(11, 55, 2)가 된다. 여기서 (11, 5, 2)가 처음에 일치하게 되므로 시퀀스 A의 i값은 0, B의 j값은 3이 되고 matchSize는 3이 된다. 이후 시퀀스 B에서 11이 등장하는 두 번째 위치가 8이므로 여기서부터 비교하기 시작한다. (11, 55, 2, 1, 85)의 matchSize가 5로 이전의 3보다 크다. 따라서 이전의 (11, 5, 2)는 후보에서 탈락하게 된다.

그림 4. 가장 긴 서브 시퀀스 찾기
Fig. 4. Finding of MaxMatchString

둘째, 시퀀스 A, B에서 일치된 서브 시퀀스를 마크하는 단계이다. 이 단계는 이전에 찾은 서브 시퀀스를 다시 검사하지 않도록 마크하는 것이 목적이다.

셋째, 시퀀스 A, B에서 minMatchLength 이상의 일치가 발생할 때까지 위의 두 단계를 반복한다. minMatchLength 보다 작은 일치만 발생하면 서브시퀀스 찾는 단계를 종료하고 유사도 값을 계산하여 반환한다. 그림 4를 예로 하면 {11, 55, 2, 1, 85}를 가지고 있고 Length(A)는 16이다.

$$sim(A,B) = (2 * \frac{totalMatchSize}{Length(A) + Length(B)}) \quad (6)$$

식 (6)은 다이스 상관계수로서 시퀀스 A, B의 일치하는 공통 서브 시퀀스들의 찾아 유사도를 판별한다. 본 논문에서와 같이 두 문서의 공통적인 서브 시퀀스를 찾아 계산하는 방법에 매우 적합한 방법이기 때문이다. 알고리즘에서 유사도를 계산하는 방법은 먼저 비교 대상이 되는 문서 A, B의 각각의 유사도 퍼센트를 구한다. 그런 다음 계산된 문서 A, B의 유사도 차이가 0.1을 넘지 않는 범위 내에 있다면 식 (6)을 이용하여 계산한다. 만약 0.1를 넘는다면 문서 A, B에서 유사도 가 높은 값을 전체 유사도로 평가한다. 여기서 0.1는 유사도를 평가하는데 가장 많이 사용되는 평균 절대오차(mean absolute error) 방법을 사용하여 나온 값이다. 평균 절대오차란 실제 값과 예측 값의 절대 값 차이의 합을 예측기간의 수로 나눈 값을 말한다. 본 논문에서 사용한 평균 절대 오차는 문서 A가 B에 대한 유사도 X값과 B가 A에 대한 유사도 Y값의 평균값 Z를 예측 값으로 계산하고 식(6)을 이용하여 계산한 결과 값을 실제 값으로 계산하였다.

Ⅳ. 결론

일반적으로 프로그램 언어와는 달리 자연어로 된 문서들은 구조적 정보를 가지고 있지 않다. 기존에는 논문과 같이 자연어로 작성된 문서들의 유사도는 통계적인 방법, 즉 지문법을 이용하여 유

사도를 측정하였다. 지문법을 이용한 유사도 측정 방법은 문서의 크기와 관계없이 빠르게 유사도를 측정할 수 있는 장점이 있다. 그러나 문서의 부분적인 유사도, 즉 문장과 문장, 문단과 문단간의 유사도를 측정할 수 없고 전체 유사도만 평가할 수 있는 단점이 있다. 또한 본 논문의 실험에서와 같이 한쪽 논문의 크기가 커지게 되면 매우 낮은 유사도 결과를 보이는 단점이 있음을 보였다. 이러한 통계적 분석 방법의 단점을 극복하기 위해서 본 논문에서는 논문마다 제시된 특별한 양식을 구조적인 형태로 나타내기 위해서 XML 스키마를 이용하여 표현 하였다. 또한 생성된 XML 스키마를 이용하여 논문을 XML문서로 변환하였고 이를 AST의 자료구조로 형태로 표현하였다. 이렇게 생성된 AST를 이용하여 구조적인 분석 방법으로 유사도 측정방법을 제시하였다. 또한 구조적인 분석 방법을 이용하면 기존의 통계적 분석 방법처럼 논문 전체만을 비교 하는 것이 아니라 논문을 구성단위별로 비교 할 수 있는 장점이 있음을 보였다. 즉 논문의 요약, 서론 ,본론 등 어떤 부분이 매우 유사한지를 알 수 있었고 이를 통해 두 논문의 관계까지도 유추할 수 있었다. 하지만 기존의 통계적 분석 방법에 비해서 속도가 느린 단점을 가지고 있다. 이러한 단점은 일치하는 패턴을 찾는 시간이 많이 걸리기 때문인데 개선해야할 부분이다. 향후 연구에는 본 시스템의 성능저하의 주원인인 패턴을 찾는 방법의 개선이 필요하다. 이러한 패턴을 빠르게 찾기 위해서 패턴의 정보를 해시 함수에 저장하여 패턴 탐색을 좀 더 빠르게 수행할 예정이다.

참고문헌

[1] 김영철, “문서와 프로그래밍 언어의 표절 검사 기술에 관한 연구”, 한국경영교육학회 제48집, 2007.

[2] 김수영, “표절과 올바른 인용 방법”, 가정의학회지, pp167-174, 2008.

[3] http://en.wikipedia.org/wiki/Distance

[4] http://en.wikipedia.org/wiki/Correlation

[5] Salton, Gerard., Introduction to Modern Information Retrieval, McGraw-Hill, 1983.

[6] C. J. van Rijsbergen, Information retrieval, Butterworths, 1979.

[7] 강승식, 권혁일, 김동렬, “한국어 자동 색인을 위한 형태소 분석 기능”, 한국정보과학회, 제22권 제1호, pp.929~932, 1995.