공통 Phrase의 관계 그래프와 Suffix Tree 문서 모델을 이용한 문서 군집화 기법

Document Clustering with Relational Graph Of Common Phrase and Suffix Tree Document Model

조윤호, 이상근
고려대학교 정보통신대학 컴퓨터통신공학부

Yoon-Ho Cho(cloudjo21@korea.ac.kr), SangKeun Lee(yalphy@korea.ac.kr)

요약

기존의 문서 군집화 기법 NSTC은 문서 군집화 과정 내에서 TF-IDF를 이용하여 문서간 유사도를 측정한다. 본 논문에서는 TF-IDF가 아닌, 공통 Phrase의 관계 그래프를 이용한 새로운 문서간 유사도 측정을 제안한다. 이 방법은 문서 집합 내의 공통 Phrase들의 관계를 나타낸 관계 그래프를 통해 공통 Phrase의 가중치를 부여하는 방법을 제시한다. 또한 실험을 통해 NSTC와 비교하여 본 논문에서 제안한 문서간 유사도 측정 기법이 문서 군집화에 더욱 효과적임을 보였다.

■ 중심어 : | 알고리즘 | 군집화 | 유사도 측정 | 문서 모델 | 관계 그래프 | Suffix Tree |

Abstract

Previous document clustering method, NSTC measures similarities between two document pairs using TF-IDF during web document clustering. In this paper, we propose new similarity measure using common phrase-based relational graph, not TF-IDF. This method suggests that weighting common phrases by relational graph presenting relationship among common phrases in document collection. And experimental results indicate that proposed method is more effective in clustering document collection than NSTC.

■ keyword : | Algorithms | Clustering | Similarity Measure | Document Model | Relational Graph | Suffix Tree |

I. 서 론

웹을 사용하는 사용자들은 단지 웹 검색엔진이 반환한 수십 개의 목록으로 이루어진 검색 결과들 속에서 관심 있는 종류의 웹 페이지를 일일이 확인하며 찾아다녀야만 했다. 하지만 웹의 발전과 더불어 사용자에게 서로 유사한 주제로 분류된 상태의 검색 결과를 제공해주는 서비스들이 발전해왔고, 이를 위한 방법 중에 하나인 군집화 하는 기법 또한 다양하게 발전하였다.

그 중에서도 NSTC(New Suffix Tree document Clustering algorithm)는 Suffix Tree와 공통 Phrase로 문서 모델을 만들고, 공통 Phrase의 TF-IDF(Term Frequency - Inverted Document Frequency)를 이용한 문서간 유사도 기법을 계층적 군집화 알고리즘에 적용

* 본 연구는 "2단계 BK21" 지원사업의 연구 결과로 수행되었습니다."
접수번호 : #081216-001
접수일자 : 2008년 12월 16일
심사완료일 : 2009년 02월 12일
교신저자 : 이상근, e-mail : yalphy@korea.ac.kr

하였다[1]. TF-IDF는 문서 집합 내에서 다수의 문서들에 걸쳐서 나타나는 단어에 대한 가중치를 낮추는 데에 효과적이다[2]. 기존의 텍스트 문서, 예를 들면 의학 분야과 같은 특정 분야에서만 통용되는 문서들은 일반적으로 하나의 주제와 관련된 단어들로 많이 표현되어 있다. 하지만 그에 비해 웹 문서는 하나의 주제에 대해 보다 자유롭게 표현하는 경향이 일부 있으며, 실제 주제에 대한 내용과 관계없는 표현들도 뒤섞여 있기 때문에 주제에 대한 단어 표현들이 대부분 단편적으로 나타난다. 따라서 기존의 텍스트 문서의 경우와는 다르게, 웹 문서에 대한 군집화는 군집화 이전의 문서간 유사도 측정 단계에 대한 변화가 필요하다. 즉, 문서간 유사도 측정 단계 내에서 구성요소(예를 들어, 단어 또는 하나 이상의 단어 집합인 Phrase)에 대한 가중치 부여 방법이 바뀌어야 됨을 뜻한다.

본 논문에서는 군집화 이전의 문서간 유사도 측정 단계에서 웹 문서에 맞는 가중치 부여 방법을 제안한다. 먼저, 공통 Phrase의 TF와 웹 문서 내의 한 블록 내에 공통 Phrase간의 출현 빈도수를 이용하여 공통 Phrase 기반 관계 그래프를 구축한다. 다음으로 이를 활용하여 공통 Phrase들의 가중치를 구하고, 문서간의 유사도를 측정하여 문서 집합에 대해 군집화를 수행한다.

II. 관련연구

Suffix tree는 하나의 문자열의 모든 접미사(suffix)를 tree형태로 만든 자료구조이다. Suffix Tree를 이용한 목적은 웹 문서 집합에서 공통적으로 나타나는 Phrase (하나 이상의 단어의 연속적인 집합)를 활용하기 위해서 이다. 하나의 문서의 주제 또는 분류를 표현하는 데에 있어서 하나 이상의 단어도 가능할 뿐더러, 그 순서도 중요한 표현 정보이기 때문에 이러한 정보를 군집화에 활용 가능하다고 본다. 본 논문에서는 E. Ukkonen의 Suffix Tree[3]를 구축했다 E. Ukkonen의 Suffix Tree는 지금까지 나온 세 종류의 Suffix Tree 중 가장 최근에 나온 것으로, Suffix Tree를 새로운 단어들의 입력에 대해 O(n)으로 구축할 수 있도록 한다. 비록 McCreight의 Suffix Tree[4]에 비해서 공간 비용의 측면에서는 효율적이지 못하지만 온라인 특성에 맞게 지속적이고 동적으로 Suffix Tree를 구축이 가능하다는 장점이 있다. Suffix Tree내에서 특정 Phrase는 다음과 같이 표현된다. 부모 노드에서 자식 노드 사이의 연결선에 하나의 Phrase가 표현될 수 있다. 이 때 루트 노드로부터 특정 자손 노드까지 경유하면서 지나는 연결선 하나하나의 Phrase들을 순서대로 합친 최종 Phrase가 특정 Phrase가 되는 것이다.

Suffix Tree는 본래 하나의 문자열 내에서의 Suffix들의 집합을 나타낸 자료구조이지만 본 논문에서는 문서 집합을 문자열이 아니라 단어의 연속체 집합으로 본다. 예를 들어 I love you의 suffix 집합은 {I love you, love you, you}와 같은 식이다. Suffix Tree의 뿌리(root) 노드와 잎새(leaf) 노드를 제외한 내부(internal) 노드는 문서 집합 내에 공통적으로 나타나는 하나 이상의 단어의 집합인 공통 Phrase를 가리키며, 어떤 하나의 Suffix의 부분에 속한다.

공통 Phrase가 군집화 이전의 문서간 유사도 측정에 줄 수 있는 장점을 살펴보면 다음과 같다. Phrase내의 단어들이 가지는 순서는 중요한 의미정보를 제공한다. 가령, “미술 치료” 라는 단어의 집합인 Phrase는 문서의 분류나 주제를 나타내는 데에 의미를 가질 수 있지만 단순히 문서 내에 나타난 “미술”과 “치료”라는 단어 자체만으로는 그 문서의 주제나 분류를 결정하기에는 부족할 수 있다.

공통 Phrase 추출은 내부 노드들 중 가장 긴 Phrase를 갖는 노드의 Suffix Link를 따라 내부 노드들을 경유하여 최종적으로 루트 노드에 도달하게 된다. 이 과정에서 방문한 내부 노드들이 웹 문서 집합 내의 공통 Phrase의 집합이다. 아래는 Suffix Tree의 간단한 예이다.

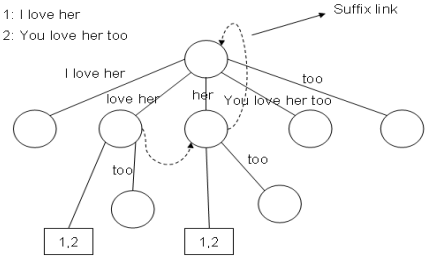

그림 1. Suffix Tree의 예

NSTC에서는 기존의 Vector 공간 문서 모델을 이용하여 군집화 단계 내에서 공통 Phrase의 TF-IDF를 이용하였다[1]. 추가로 여러 단어의 집합인 Phrase와 단어들 간의 순서에 대한 정보가 군집화에 영향을 줄 수 있음을 보여준 STC(Suffix Tree Clustering) 알고리즘의 Suffix Tree 문서 모델[5]의 조합을 통해 문서간의 유사도 측정 단계에서 새로운 방법을 제시했다. 또한 이 방법을 군집화 알고리즘에 적용 및 실험을 통해 문서 군집화에 효과적임을 보였다. 이러한 유사도 측정 방법 및 사용된 군집화 알고리즘의 전 단계를 NSTC(New Suffix Tree Document Clustering Algorithm)이라 명명하였다.

TermRank는 구글의 PageRank의 철학에 기반을 둔 관계 그래프(Relational Graph)를 생성하여 단어의 가중치 부여를 위해 제안된 알고리즘이다[6]. 이 알고리즘의 목적은 문서 집합 내에서의 카테고리를 대표하는 단어들에 대한 적절한 가중치 부여를 통해 군집화 알고리즘의 성능을 높일 수 있도록 하는 것이다. 단어들의 관계 그래프에서 초기에 각각의 노드에는 문서 집합 내의 단어의 TF (Term frequency)를 포함한다. 한 문서의 블록(. <p>, , <ul>, <ol> 등등)내에 두 단어들 간의 동시 출현 빈도수는 노드 간의 무방향성 연결선의 가중치로 구성된다. 노드 간에 무방향성 연결선으로 연결되는 이유는 다음과 같다. PageRank에서는 일반적으로 한 문서에서 다른 문서로의 하이퍼링크를 통해 연결된다. 이것은 한 방향으로의 연결만을 의미한다. 하지만 TermRank에서는 두 단어가 한 문서에 동시에 나타난다. 이 의미는 PageRank와 다르게 두 단어 간의 양방향으로 연결됨을 의미한다. 이러한 결과로 TermRank의 식은 기존 PageRank 식의 무작위 점프 부분을 나타내는 항을 제거한 형태로 수정되었다. PageRank의 Random Surfer 모델에서 Random Surfer는 문서의 링크들을 따라가다가 외향 링크(outgoing link)가 없는 문서에 도착했을 경우를 대비해 임의의 한 문서로 무작위 점프(random jump)를 할 수 있어야 되는데 이러한 필요가 없어졌기 때문이다.

이러한 노드와 연결선에 할당한 정보를 PageRank의 계산식을 사용한 근사방법을 통해 각 단어에 가중치가 부여된다. 부여된 가중치가 큰 단어들은 웹 검색 엔진에 던진 질의 키워드에 대해 관련이 깊고, 그래프 내의 단어의 가중치가 작을수록 카테고리에 대한 관련성이 떨어진다.

본 논문에서는 GAHC(Group-average Hierarchical Clustering algorithm)[7]를 군집화 알고리즘으로 이용한다. 계층적 군집화(Hierarchical Clustering)에는 크게 Bottom-up과 Top-down 두 가지 방법이 있다. 그 중에 본 논문에서는 NSTC에서 군집화 알고리즘으로 이용한 Bottom-up 방법 중 하나인 GAHC를 채택했다. Bottom-up 군집화는 첫 단계에서 모든 데이터들을 각각의 군집으로 보고, 군집 간의 유사도가 가장 높은 둘을 하나의 군집으로 합치고 새로 형성된 군집들 간의 유사도를 계산한다. 이러한 과정을 하나의 수형도(dendrogram)가 만들어질 때까지 즉, 하나의 군집으로 합쳐질 때까지 반복한다. GAHC는 이러한 Bottom-up 방법을 기반으로 군집 내의 데이터들 간의 평균 유사도를 이용하여 군집화를 수행해 나간다.

III. 문서 군집화 과정

본 논문에서는 TermRank 알고리즘[6]에서 제시했던 단어 관계 그래프가 아닌 Phrase 관계 그래프를 이용한다. [그림 2]은 본 논문과 NSTC의 두 공통 Phrase 가중치 부여 방법을 포함한 전체 군집화 과정에 대한 흐름도이다. 첫 번째 단계에서는 문서 집합(document collection)을 Suffix tree 문서 모델로 구축한다. 구축된 Suffix tree 모델에서 내부 노드만을 순회하여 문서 집합 내에서 두 번 이상 공통적으로 나타나는 공통 Phrase들을 추출해낸다. 두 번째 단계에서는 Phrase에 가중치를 부여한다. 이 때 NSTC에서는 공통 Phrase 집합을 벡터 공간 문서 모델(Vector space document model)로 표현하여 각 Phrase별 TF-IDF를 계산한 가중치를 부여한다[1]. 본 논문에서는 공통 Phrase 집합을 관계 그래프 문서 모델로 표현하고, 각 Phrase의 가중치는 1) 해당 Phrase의 연결선들의 가중치, 2) Phrase의 이웃 Phrase 노드들의 가중치, 그리고 3) 이웃

Phrase 노드들의 연결선들의 가중치를 이용한 근사 방법(approximation method)을 통해 계산 및 부여된다. 세 번째 단계에서는 공통 Phrase의 가중치들로 표현된 문서 벡터 쌍마다 유사도를 계산한다. 두 문서간의 유사도가 높을수록 두 문서가 하나의 군집이나 분류에 속할 가능성이 높음을 의미한다. 이 때 유사도 측정 시에 사용되는 방법은 문서 군집화에 널리 쓰이는 Cosine 유사도 측정 방법을 사용하였다. 하나의 문서 d_i와 다른 문서 d_j의 간의 유사도는 아래의 식 (1)과 같이 정해진다.

$$\cos(\vec{d_i}, \vec{d_j}) = \frac{\vec{d_i} \bullet \vec{d_j}}{|\vec{d_i}| \times |\vec{d_j}|} \quad (1)$$

앞서 언급했듯이, 하나의 문서를 대표하는 문서 벡터 내의 한 요소는 해당 공통 Phrase의 가중치이다. 그리고 문서 벡터에는 수많은 공통 Phrase들의 가중치들로 구성되어 있다. 문서 d_i의 문서 벡터와 문서 d_j의 문서 벡터를 이루는 공통 Phrase들의 가중치들이 서로 비슷한 크기를 가질수록 두 벡터 간의 Cosine 값(유사도)이 커진다는 특징을 이용한 유사도 측정 방법이다.

마지막 네 번째 단계로 Bottom-up 방식의 계층적 군집화 알고리즘인 GAHC(Group-average Agglomerative Hierarchical Clustering algorithm)을 통해 이전 단계의 결과인 문서간 유사도 쌍들을 비교해 나가면서 하나의 군집으로 모일 때까지 계층 군집화를 수행한다[7]. 이 때 군집간의 유사도와 두 군집을 합쳤을 때의 개수를 이용한 평균 유사도를 합쳐진 군집의 유사도로 한다. 마지막 결과는 k개의 적절한 개수의 문서를 가진 군집으로 구성된다. k는 실험에서 사용되는 데이터 집합 내 그룹의 개수로 실험 전에 미리 제공된다.

Suffix Tree에서 공통 Phrase를 검색하여 추출하는 데에 드는 시간 복잡도는 다음과 같다. 문서 집합 내의 문서의 개수를 N, 한 문서 내 단어의 평균 개수를 m로 한 경우, Nm의 복잡도를 보여준다. 왜냐하면 suffix link를 따라 공통 Phrase를 추출하기 때문이다. p를 Suffix Tree로 추출한 문서 집합 내 공통 Phrase의 개수로 두고, Cosine 유사도 측정시의 시간 복잡도는 $2p$에 선형적으로 비례하여 증가한다. 마지막으로 문서 집합간의 모든 pairwise 유사도를 계산할 때에는 $N(N-1)p = pN^2$의 시간 복잡도를 가진다.

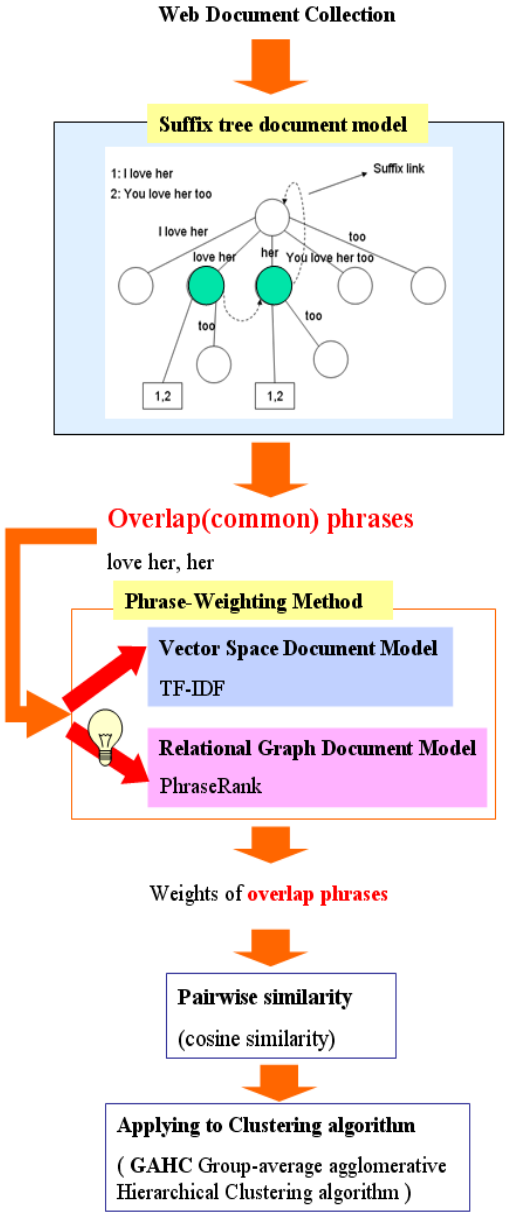

그림 2. 제안된 기법을 포함한 군집화 과정

IV. PhraseRank 알고리즘

[그림 3]은 PhraseRank 알고리즘을 기술한다. 제시한 알고리즘을 세부적으로 설명하기 이전에 PhraseRank 전체 단계를 간략히 설명한다.

```
D: a document collection
G: the relational graph of common phrase
CreateRelGraph(P, I)
  P: a list of common phrase
  I: inverted barrel containing inverted indices of words
  Let p_i denote the ith phrase of P.
  Set G := ∅.
  for each phrase p_i ∈ P do
    Let TF_i denote the term frequency of p_i.
    Let G.node_i denote the weight of one node stood for phrase.
    Get TF_i of p_i using I.
    G.node_i := TF_i.
  end
  for each document d_i ∈ D do
    Let (p_k, p_l) denote the all pairs for distinct two phrases in P
    for each (p_k, p_l) ∈ P do
      Let CO_j denote the co-occurrence count between (p_k, p_l).
      Let G.edge_j denote the weight of edge j.
      Get CO_j of (p_k, p_l).
      G.edge_j := CO_j.
    end
  end
return G

All-PhraseRank(N, δ)
  N: the limit number calculating phraserank enough
  δ: a small number for approximation
  Let PhR denote the phraserank set of all common phrases.
  Let t denote the current step of phraserank calculation.
  Let b denote the flag to decide stop phraserank calculation.
  Set t := 0.
  Set b := 1.
  while t < N AND b = 1 do
    Let G.V denote the vertex set of G.
    Let v_i denote the weight of vertex represented phrase.
    b := 1.
    for each vertex v_i ∈ G.V do
      Let calcPhraseRank denote the formula in IV.2
      PhR^(t+1) := calcPhraseRank(v_i , G).
      for each vertex v_j ∈ G.V do
        if δ < PhR^(t+1)(v_j)-PhR^(t)(v_j) then
          b := 0.
          escape this for each.
        end
      end
      if b = 1 then
        escape while.
      t := t+1.
    end
    PhR := PhR^(t).
return PhR
```

그림 3. PhraseRank 알고리즘

1. 공통 Phrase의 관계 그래프 생성

TermRank의 단어 관계 그래프에서는 하나의 단어와 하나의 블록 내에서 동시에 일어나는 단어들 간의 정보를 이용하여 그래프를 구성했다. 하지만 본 논문에서는 문서 집합의 공통 Phrase의 정보와 공통 Phrase간의 정보를 이용하여 관계 그래프를 만든다. 즉, 공통 Phrase의 관계 그래프에서의 노드는 Phrase의 가중치를 포함하고, 연결선의 가중치는 TermRank와는 달리, 한 문서 내 두 Phrase쌍들 간의 동시 출현 빈도수로 구성된다. 이러한 Phrase들의 정보는 구글 검색 엔진에서 사용한 역 인덱스 리스트[8]의 구현을 이용하여 추출해낸다. 아래 그림은 본 논문의 관계 그래프의 기본적인 형태이다.

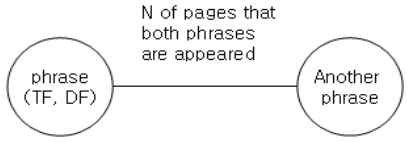

그림 4. 공통 Phrase의 관계 그래프의 기본 형태

공통 Phrase의 관계 그래프에 담긴 기본적인 아이디어는 다음과 같다. 어떤 한 Phrase가 다른 Phrase간의 연결선의 가중치가 높고 연결선의 수가 적을수록 해당 Phrase는 문서 분류에 영향을 미칠 가능성이 높은 중요한 Phrase다. 그러나 만약 Phrase가 다른 Phrase간의 연결선의 가중치가 낮고 연결선의 수가 많을수록 해당 Phrase는 문서의 분류와 크게 상관없이 흔히 쓰이는 Phrase로 본다.

PhraseRank의 공통 Phrase의 관계 그래프와 TermRank의 단어 관계 그래프의 결정적 차이는 다음과 같다. 첫 번째, 공통 Phrase의 관계 그래프에서 한 노드는 공통 Phrase 하나의 가중치를 나타내고 단어 관계 그래프에의 하나의 노드는 단어 하나의 가중치를 나타낸다. Phrase는 하나 이상의 단어들의 집합이다. 공통Phrase는 문서 집합에 대해 공통적으로 나타나는 둘 이상의 단어집합 뿐만 아니라 공통 단어도 포함한다. TermRank의 단어 관계 그래프에서 단지 문서에 나오는 모든 단어들에 대해 관계 그래프를 만든 것에 비해, PhraseRank의 공통 Phrase의 관계 그래프에서는 공통으로 나오는 Phrase(단어 포함)만을 다루기 때문에 분

류에 도움이 되지 않는 일반적인 단어들을 배제할 수 있다. 두 번째, 연결선의 형성이 다르다. 공통 Phrase의 관계 그래프에서는 한 문서 내에서 두 Phrase가 동시에 일어나는 경우에 가중치가 더해진다. 이와 다르게, 단어 관계 그래프에서는 <p>태그와 같은 문단이나 한 구획 내에서 동시에 일어나는 두 단어가 있을 때 연결선의 가중치가 더해진다. 단어 관계 그래프에서는 문서 내 모든 단어들에 대해 만들기 때문에 두 단어가 동시에 나오는 횟수를 세기 위한 단위가 문단이나 구획이 적절할 수 있다. 하지만 공통 Phrase의 관계 그래프에는 문서 집합 내에서 공통적으로 나오는 Phrase만을 다루기 때문에 단어의 출현 횟수보다 나오는 횟수가 크게 적다. 따라서 Phrase 둘이 동시에 나오는 횟수를 세기 위한 단위를 문단이나 구획으로 나누는 것은 적절하지 못하기 때문에 하나의 문서를 단위로 잡아 연결선의 가중치를 구하도록 하였다.

2. Phrase의 가중치 계산

이 단계에서는 IV.1.의 단계에서 구축한 공통 Phrase 기반 관계 그래프를 이용해 Phrase의 가중치를 계산한다.

PageRank에서는 사용자가 특정 주제를 가지고 웹을 돌아다닌다. 그러므로 어떤 페이지에 있는 여러 링크 중 방문한 목적과 관련 있는 링크를 클릭할 확률이 높다[9]. 이러한 PageRank의 철학에 기반을 둔 TermRank에서는, 어떤 단어는 여러 연결선 중 가중치가 높은 연결선을 따라가면 문서의 분류와 관계가 높은 단어를 찾아갈 확률이 높아진다. PageRank와 TermRank의 차이는 PageRank는 방향성을 갖고 가중치가 없는 연결선이 포함된 그래프이고 TermRank는 무방향성에 가중치가 있는 연결선이 포함된 그래프라는 것이다. PhraseRank는 TermRank와는 다르게 하나의 단어가 아닌 공통 Phrase의 가중치를 구한다. 아래는 PhraseRank를 계산하는 기본 수식이다.

$$PhR(i) = \sum_{j \in N(i)} \frac{PhR(j) \cdot w_{ij}}{\sum_{k \in N(j)} w_{jk}} \quad (2)$$

$$PhR^{(0)}(i) = \frac{w_i}{\sum_{j \in V(G)} w_j} = TF(i) \quad (3)$$

$$PhR^{(0)}(i) = \frac{w_i}{\sum_{j \in V(G)} w_j} = TF(i) \quad (4)$$

식 (2)은 PhraseRank의 기본 수식이다. *PhR(i)*은 *phrase i*의 가중치, 즉 노드 *i*의 가중치를 말한다. *N(i)*는 노드 *i*의 이웃 노드들의 집합을, w_{ij}는 노드 *i*와 노드 *j* 사이의 연결선의 가중치를 나타낸다. 노드 *i*의 이웃 노드 *j*들로의 연결선의 가중치와 이웃 노드 *j*들의 가중치가 높으면서, 이웃 노드 *j*와 이웃 노드 *k*들로의 연결선의 가중치가 낮으면서 개수가 작으면 노드 *i*의 가중치가 높아진다. 즉, 전체 문서에 걸쳐 동시에 나타나는 공통 phrase들이 관계 그래프 내에서 각각의 그룹을 이루면서 그룹 내에 공통 Phrase가 다수 있으면 서로간의 가중치를 높여줄 수 있다. 이는 곧, 하나의 카테고리를 대표하는 공통 phrase들이 부각되는 효과를 내게 된다.

식 (3)에서의 노드 *i*의 즉, PhraseRank의 0 단계는 아직 한 노드에서 다른 노드들로의 연결선이 없는 상태이다. 따라서 공통 Phrase의 가중치(*PhR(i)*)를 전체 문서 집합 내에서의 Phrase의 TF인 출현빈도(*TF(i)*)로 초기화시킨다. w_i는 연결선이 없기 때문에 연결선이 아닌 노드 *i* 자신의 가중치를 나타내기 때문이다. 또한, 연결선이 없는 상태에서 한 노드의 이웃은 그래프 내 전체 노드들로 본다. *V(G)*는 그래프 내의 노드(Vertex)의 집합을 가리킨다.

이후에 식 (4)과 같이 근사방법(approximation method)을 사용하여 t 단계 Phrase의 가중치와 t+1 단계의 Phrase의 가중치의 차이가 특정 근사치에 도달할 때까지 반복하여 계산한다.

PhraseRank 계산 시의 시간 복잡도는, 노드의 개수는 공통 Phrase의 개수와 동일하기 때문에 *p*, 하나의 노드가 갖는 연결선의 평균 개수를 *e*, 그리고 *k*를 PhraseRank 계산 반복 횟수를 나타낸다고 가정할 때, $O(k(p+e^2))$의 복잡도를 가진다. 노드의 가중치 및 연결선의 이웃들을 접근하고 할당하는 횟수를 기준으로 계산 시간 복잡도를 분석하였다. 또한 관계 그래프의 구현이 노드별로 연결선의 리스트를 갖기 때문에 공통 Phrase의 관계 그래프의 공간 복잡도는 $2pe$와 같다.

3. 단절된 공통 Phrase의 관계 그래프 문제 및 해결

기존 TermRank[6]에서는 문서 집합 내에 나타나는 단어들에 대해 가중치 부여 및 순위화하고, 문서간의 pairwise 유사도 측정 이후 군집화 알고리즘에 적용시켰다. 하지만 계층 군집화 과정에서는 한 가지 예외가 일어날 수 있다. 대개, 어떤 한 문서에 나온 단어는 다른 문서에서도 나올 수 있고, 특정 한 문서에만 나오는 단어가 있다고 하더라도 다른 문서에 일반적으로 나오는 단어도 같이 포함하고 있는 경우가 대부분이다. 하지만 어떤 한 문서의 경우 다른 문서에는 나타나지 않는 단어들로만 구성되어 있는 경우가 있다. 즉, 해당 문서의 단어들로만 구성된 그래프가 다른 문서들의 단어들로 구성된 그래프와는 연결되지 않는 상태가 된다. 이러한 경우 해당 문서와 다른 문서들 간의 유사도는 모두 0의 값을 가지게 된다. 왜냐하면 단어들의 관계 그래프에서 해당 문서의 단어는 다른 문서의 단어들과의 어떠한 연결선도 갖지 못하기 때문이다.

이 문제는 단어가 아닌 Phrase의 경우로 바뀌도 동일하다. 어떤 한 문서가 다른 문서와의 유사도가 0이라는 것은 계층 군집화 과정에서 어떠한 군집에 속할 수 있는지에 대한 기준이 없기 때문에 해당 문서의 군집을 결정할 수가 없다.

이를 해결하기 위해 문서를 표현하는 문서 벡터 내의 가중치가 0인 Phrase들의 가중치를 원래의 문서간 유사도에 거의 영향을 미치지 못할 정도로 극히 작은 특정 값으로 새로 할당한다. 이를 통해 문서간 유사도 측정 단계에서 문서간의 유사도가 0인 문서 쌍이 발생하지 않기 때문에 계층적 군집화 알고리즘을 수행하여도 특정 군집에 속할 수 있게 된다. NSTC에서는 TF-IDF 식에서 Phrase의 가중치가 항상 0이 나올 수 없도록 보장하고 있기 때문에 이러한 문제가 일어나지 않는다.

V. 실험 및 결과

본 논문에서는 NSTC의 실험에 이용되었던 두 가지 문서 Corpus에 대해서 실험을 수행하였다.

OHSUMED(Oregon Health Sciences University MEDLINE database)[10]은 심혈관계 질환을 앓고 있는 환자와 세부 질병이 기록된 정보 검색 연구 목적용 문서 Corpus(1987-1991)으로 색인이 달린(indexing) 용어들과 요약 등을 포함한다. RCV1 (Reuters Corpus Volume 1)[11]는 Reuters의 1년 동안의 뉴스(1996-08-20~1997-08-19)를 담은 연구 목적용 문서 Corpus이다.

OHSUMED는 총 381MB, 약 29만개의 문서로 이루어져 있다. 이 중에서 총 171KB, 8개의 그룹(MeSH 인덱스 단어: MSH1058, MSH1262, MSH1473, MSH1486, MSH1713, MSH2025, MSH2030, MSH2235)에서 무작위로 100개씩 추출, 총 800개의 문서를 실험 문서 Corpus 중 하나로 선별하였다. RCV1은 총 2.3GB, 약 80만개의 뉴스 기사로 구성되어 있고, 10개의 주제별 카테고리(카테고리 인덱스 단어: C11, C12, C21, C41, E11, GREL, GSCI, GSPO, GWEA, M11)에 속하는 총 237MB인 약 15만개의 기사 내의 텍스트를 추출하였다. 그 중에서 다시 주제 하나당 무작위로 선택한 200개의 문서를 뽑아 총 2000개의 문서를 선별하였다.

문서 집합은 OHSUMED와 RCV1으로부터 각각 3가지 종류로 구성하였다. 앞서 추출한 OHSUMED로부터 MSH2235, MSH2025, MSH2030 3개의 그룹으로 문서 집합-1을, MSH1058, MSH1262, MSH1473, MSH1486, MSH1713 5개의 그룹으로 를 포함한 문서 집합-2를, 마지막으로 8개 그룹을 모두 포함한 문서 집합-3을 구성하였다. RCV1 또한 앞에서 추출한 문서로부터 세 문서 집합을 구성하였다. 문서 집합-4는 GREL, GSCI, GSPO, GWEA 4개의 카테고리에 해당하는 문서 100개씩 400개로, 문서 집합-5는 나머지 6개의 카테고리 해당 문서 100개씩 600개, 마지막으로 문서 집합-6은 10개 카테고리 총 2000개의 문서로 구성하였다. 아래는 문서 집합을 나타내는 표이다.

표 1. 문서 집합 개요 (O: OHSUMED, R: RCV1)

문서 집합	DS1	DS2	DS3	DS4	DS5	DS6
Corpus	O	O	O	R	R	R
카테고리 수	3	5	8	4	6	10
문서 수	300	500	800	400	600	2000

본 논문에서 사용된 실험 환경은 다음과 같다. H/W는 2.33GHz의 Intel Core2 Duo CPU, 2GB RAM을, 사용된 OS는 Microsoft Windows XP이다. 논문에서 사용된 S/W 모듈들은 참고 논문에서 설명하는 배경지식을 통해 직접 구현되었다.

성능 평가는 *F-measure*를 사용하여 군집화 결과를 평가하였다. *Precision*과 *Recall*의 조화 평균인 *F-measure*는 군집화나 분류화(classification)를 평가할 때 널리 쓰이는 성능 측정 방법이다[12].

$$p_{ij}=\frac{n_{ij}}{n_j}, r_{ij}=\frac{n_{ij}}{n_i}, i=1,...,l, j=1,...,k \quad (5)$$

$$F_{ij}=\frac{2p_{ij}r_{ij}}{p_{ij}+r_{ij}} \quad (6)$$

$$F:=\sum_{i=1}^{l}\frac{n_i}{n}max_j\{F_{ij}\}, j=1,...,k \quad (7)$$

n_{ij}는 카테고리 i와 군집 j에 동시에 속하는 문서 수, n_i는 카테고리 i에 속하는 문서 수, n_j는 군집 j에 속하는 문서 수를 가리킨다. p_{ij}는 카테고리 i와 군집 j에 속하는 문서들의 *Precision*을, r_{ij}는 *Recall*을 나타낸다. F_{ij}는 카테고리 i와 군집 j에 동시에 속하는 문서들의 *F-measure*를, F는 해당 문서 집합 전체의 *F-measure*를 가리킨다.

아래 [표 2]는 NSTC와 본 논문에서 제안된 유사도 측정 기법인 PhraseRank를 포함한 군집화 알고리즘을 각각의 문서집합에 대해 *F-measure* 성능 평가를 수행한 결과이다.

표 2. 군집화 성능 측정 결과 (F-measure Score)

문서 집합	DS1	DS2	DS3	DS4	DS5	DS6	평균
NSTC	0.85	0.86	0.73	0.76	0.77	0.59	0.76
제안 기법	0.87	0.84	0.78	0.82	0.80	0.68	0.80

OHSUMED는 의학 전문 용어를 담고 있는 그 특징상 문서 내의 특정 단어들이 어떤 군집이나 분류에 속할지에 대해 뚜렷하다. 하지만 RCV1는 하나 이상의 주제를 포괄적으로 설명할 수 있는 단어를 내포하고 있기 때문에 전 처리된 실제 웹 문서의 특징을 잘 담고 있는 문서 집합이라 할 수 있다. 이와 같은 문서 집합의 특징에 맞게 PhraseRank를 적용시킨 군집화 알고리즘이 NSTC에 비해 최소 2%에서 최대 9% 정도의 성능 개선이 되었다. 기존의 일반 텍스트 문서의 가중치 부여를 위해 흔히 사용된 TF-IDF되었다. 하지만 PhraseRank에서 제시한 방법을 통해, 카테고리를 대표할 수 있는 공통 Phrase의 가중치를 관계 그래프를 통해 더욱 부각시킬 수 있었기 때문에 성능 향상을 가져올 수 있었다. 문서 집합-6(DS6)과 같은 경우, 카테고리 수가 많을 뿐만 아니라 문서의 수 또한 많다. 따라서 문서들이 포함한 단어들 속에 카테고리와 직접 관련이 없는 단어들 또한 많아졌기 때문에 문서간 유사도 측정 및 군집화의 정확도가 떨어지게 되었다. 하지만 본 방법을 통해 군집화 성능 감소폭을 소폭 완화시킬 수 있었다.

VI. 결론 및 향후 연구

본 논문에서는 군집화 알고리즘 이전 단계에서 Suffix Tree 문서 모델과 공통 Phrase기반 관계 그래프를 이용하여 문서간의 유사도를 측정하는 기법을 제안하였다.

NSTC에서의 문서간 유사도 측정 기법은 단순히 공통 Phrase의 출현 빈도를 기반으로 한 공통 Phrase 순위화를 위해 TF-IDF를 적용시켰다. 하지만 이 기법은 여러 단어나 Phrase들이 한 문서 내에 함께 출현하는 경우에도 문서에 대한 군집화나 분류에 영향을 미칠 수 있다는 점을 간과하였다.

반면에 PhraseRank는 기존의 텍스트 문서간의 유사도 측정하는 데에 사용되는 TF-IDF가 아닌 웹 문서간의 유사도 측정에 적합하도록 공통 Phrase의 관계 그래프를 구성한다. 웹 문서는 문서의 정확한 분류와 상관없이 자유롭게 표현되는 단어들이 혼재하기 때문에 문서 분류에 대한 단어 표현들이 대부분 단편적으로 나타난다. 따라서 웹 문서내의 공통 Phrase들에 대한 가중치 부여에 있어서 PhraseRank가 차별적인 방법을 제공한다.

향후 연구로는 제안된 방법의 시간 및 공간 복잡도

분석을 통해 효율적으로 알고리즘을 수행해 나갈 수 있도록 최적화시키는 것이다. 또한 단절된 공통 Phrase기반 관계 그래프 문제를 WordNet[13]을 이용해서 연결되지 못한 단어들에 대해 유사어 교환이 자동적으로 일어나도록 수정하여 대처하도록 하는 방법 또한 제시될 수 있을 것이다.

참 고 문 헌

[1] H. Chim and X. Deng, "A New Suffix Tree Similarity Measure for Document Clustering," In Proceedings of the 16th International Conference on World Wide Web, pp.121-130, 2007.

[2] G. Salton and C. Buckley, "Term-Weighting Approaches In Automatic Text Retrieval," Information Processing and Management, Vol.24, No.5, pp.513-523, 1988.

[3] E. Ukkonen, "On-Line Construction of Suffix Trees," Algorithmica, Vol.14, No.3, pp.249-260, 1995.

[4] E. M. McCreight, "A Space-Economical Suffix Tree Construction Algorithm," Journal of the ACM, Vol.23, No.2, pp.262-272, 1976.

[5] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibility Demonstration," In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp.46-54, 1998.

[6] F. Gelgi, H. Davulcu, and S. Vadrevu, "Term Ranking for Clustering Web Search Results," In Proceedings of the 10th International Workshop on Web and Database, 2007.

[7] E. M. Voorhees, "Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval," Information Processing and Management, Vol.22, No.6, pp.465-476, 1986.

[8] S. Brin and L. Page, "The Anatomy of a Large Scale Hypertextual Web Search Engine," In Proceedings of the 7th International Conference on World Wide Web, pp.107-117, 1998.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank Citation Ranking: Bringing Order to the Web," Technical Report, Stanford Digital Library Technologies Project, 1998.

[10] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam, "OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research," In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.192-201, 1994.

[11] D. D. Lewis, Y. Yang, T .G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," Journal of Machine Learning Research, Vol.5, pp.361-397, 2004.

[12] M. Rosell, V. Kann, and J. E. Litton, "Comparing comparisons: Document clustering evaluation using two manual classifications," In Proceedings of the 3th International Conference on Natural Language Processing, 2004.

[13] http://en.wikipedia.org/wiki/WordNet

저 자 소 개

조 윤 호(Yoon-Ho Cho) 준회원

- 2008년 2월 : 경희대학교 컴퓨터공학과(공학사)
- 2008년 3월 ~ 현재 : 고려대학교 컴퓨터·전파통신공학과(공학석사)

<관심분야> : 데이터마이닝, 웹 검색

이 상 근(SangKeun Lee) 정회원

- 1994년 2월 : 고려대학교 컴퓨터 공학과(공학사)
- 1996년 2월 : 고려대학교 컴퓨터 공학과(공학석사)
- 1999년 2월 : 고려대학교 컴퓨터 공학과(공학박사)
- 2000년 ~ 2001년 : Univ. of Tokyo 특별방문 연구원
- 2003년 ~ 현재 : 고려대학교 컴퓨터·전파통신공학과 조교수

<관심분야> : 웹 데이터 관리, 모바일 데이터 관리, 위치기반 정보시스템