계층적 결합형 문서 클러스터링 시스템과 복합명사 색인방법과의 연관관계 연구

The Experimental Study on the Relationship between Hierarchical Agglomerative Clustering and Compound Nouns Indexing

조 현 양(Hyun-Yang Cho)*
최 성 필(Sung-Pil Choi)**

목 차

초 록

본 논문에서는 복합명사에 대한 색인 방법을 다각적으로 적용하여 계층적 결합 문서 클러스터링 시스템의 결과를 분석한다. 우선 한글 색인 엔진과 HAC(Hierarchical Agglomerative Clustering) 엔진에 대해서 설명하고 한글 색인 엔진에서 제공되는 3가지 복합명사 분석 모드에 대해서 기술한다. 또한 구현된 클러스터링 엔진의 특징과 속도 향상을 위한 기법 등을 예시한다. 실험에서는 3가지 복합명사 색인 방법을 기준으로 문서 클러스터링을 수행하고, 실험 결과에 대한 분석에서 복합명사에 대한 색인 방법이 문서 클러스터링의 결과에 직접적인 영향을 준다는 것을 보여준다.

ABSTRACT

In this paper, we present that the result of document clustering can change dramatically with respect to the different ways of indexing compound nouns. First of all, the automatic indexing engine specialized for Korean words analysis, which also serves as the backbone engine for automatic document clustering system, is introduced. Then, the details of hierarchical agglomerative clustering(HAC) method, one of the widely used clustering methodologies in these days, was illustrated. As the result of observing the experiments, carried out in the final part of this paper, it comes to the conclusion that the various modes of indexing compound nouns have an effect on the outcome of HAC.

키워드: 자동색인, 문서클러스터링, 한글 형태소 분석
Automatic Indexing, Document Clustering, Korean Morphological Analysis

---

* 경기대학교 문헌정보학과 조교수(hycho1180@yahoo.co.kr)
** 한국과학기술정보연구원 연구원(spchoi@kisti.re.kr)
논문접수일자 2004년 11월 20일
게재확정일자 2004년 12월 16일

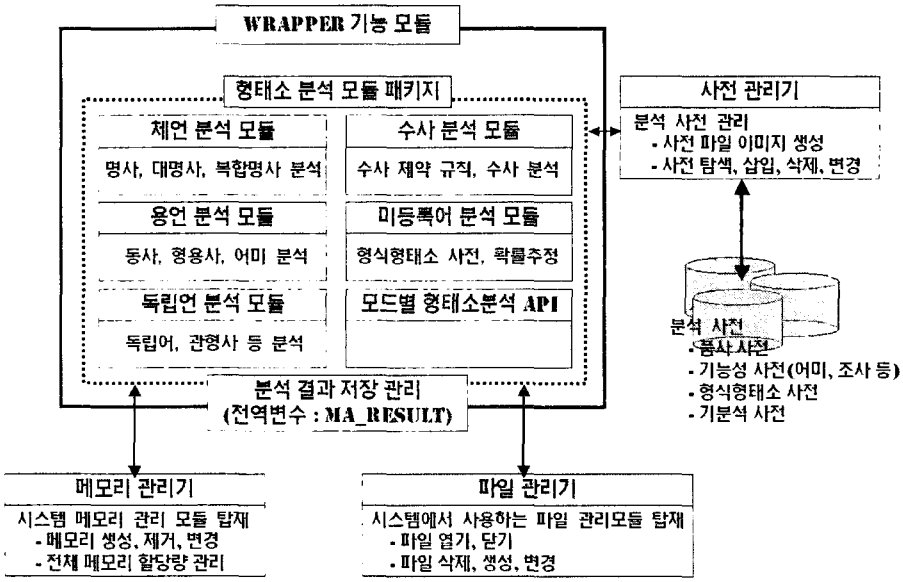
1. 서 론

최근 들어, 전자매체의 급격한 급증으로 인해 이를 관리하고 서비스하는 방법론에 대한 문제가 빈번하게 대두되고 있다. 전통적인 정보검색의 관점에서 보면 이러한 대용량 전자문서집합은 단순히 사용자 질의에 대한 매칭 대상일 수가 있지만, 사용자의 요구사항이 다양해지고 관리의 효율성이 강조됨에 따라 문서집합의 효과적인 전처리 및 구조화 등이 매우 중요한 요소로서 부각되고 있다. 이러한 가운데 서로 관련이 있거나 유사한 문서들을 자동으로 군집화하여 제시하는 자동문서 클러스터링 시스템이 현재 각광을 받고 있다. 문서 클러스터링이란 비슷한 성향을 나타내는 문서들 간의 관계를 분석하여 하나의 요소집합으로 모으는 것이다(Manning, Christopher D. and Hinrich Schutze 1999). 문서들은 특정한 기준이나 관계에 따라 타 문서와 구별되거나 서로 유사한 것끼리 집단화 될 수 있다. 문서들 간의 관계 중 대표적인 요소가 문서 유사도이며, 문서에 대한 유사성 판단을 위해서는 문서를 색인하여 색인 어휘를 추출하고 추출된 색인어를 바탕으로 문서 벡터를 구성해야 한다. 여기서 문서 벡터란 문서를 구성하는 어휘들 중 그 문서를 가장 효과적으로 특징지을 수 있는 어휘를 선택하여 이를 색인어로 구성하고 각 색인어에 대한 정보(출현빈도, 출현위치, 가중치 등)와 함께 표현한 문서별 집합을 뜻한다. 특수 목적을 위한 시스템을 제외한 대부분의 시스템들은 명사 및 명사 상당어구를 색인어로 추출하고 이를 문서 벡터로 등재시킨다.

본 논문에서는 색인어로서 명사를 추출할 때, 특히 복합명사에 대한 분석 방법이 자동문서 클러스터링의 결과에 어떠한 영향을 미치는가에 대해서 논하고자 한다. 이를 위해서 우선 2장에서는 본 시스템에서 사용된 한글 자동 색인 시스템에 대해서 설명하고, 제공되는 세 가지 복합명사 분석 방법에 대해서 설명한다. 3장에서는 이 논문에서 구현된 계층적 결합 클러스터링(Hierarchical Agglomerative Clustering, HAC) 엔진에 대해서 설명하고 그리고 마지막 4장에서는 다양한 실험을 통해서 세 가지 복합명사 분석 방법이 결과로 생성된 문서 클러스터의 특징에 어떠한 영향을 주는 가를 분석한다.

2. 한국어 자동 색인 시스템

본 논문에서 사용한 형태소 분석 시스템의 전체적인 구조는 〈그림 1〉과 같다(조현양, 최성필 2001). 〈그림 1〉에서 보는 바와 같이 각 기능적 요소들은 완벽하게 모듈화 되어 있다. 또한 시스템의 각 모듈별로 특수 기능을 수행하는 하부 모듈이 존재한다. 〈그림 1〉에서 나타나는 각 모듈은 형태소 분석 상에서 가장 크게 그 기능을 차지하는 모듈이다. 이 시스템의 장점은 만일 시스템 상에서 새로운 기능을 추가할 필요가 있을 경우 그 기능에 부합하는 모듈을 구현하여 형태소 분석 모듈 패키지의 일부분으로 추가하고 기존의 다른 모듈과 연동하는 과정만을 기술하여 적용시키면 된다는 것이다. 기능 추가를 위한 작업이 용이한 이유는 각 모듈간의 통신이 형태소 분석 결과를 저장하는 전역변수 1개와 함수 파라메터, 리턴 값



〈그림 1〉 자동 색인 시스템의 전체 구조도

으로만 이루어지기 때문이다. 형태소 분석 모듈 패키지를 둘러싸는 'WRAPPER 기능 모듈'은 사용자의 요구사항이나 시스템 적용 환경에 맞도록 형태소 분석 모듈들을 이용하여 다양한 기능을 구현하는 기능을 수행한다. 사전 관리기는 Tree 구조를 기반으로 시스템에서 사용하는 사전들을 저장하고 이를 고속으로 접근할 수 있는 API를 제공하며, 실제 어절 분석 모듈인 어절 형태소 분석기는 한국어 어절의 구성 형태에 따른 유한 오토마타의 각 상태 전의 구조로 이루어진다.

일반적인 복합명사에 대한 색인 방법은 복합명사를 무조건 단위명사로 나누는 방법이다. 그러나 복합명사 그 자체가 하나의 의미를 가지면서 단위명사로 분리되면 그 의미를 상실하는 경우가 자주 발생한다. 그 예로 "대우전자"는 "대우"와 "전자"로 분리되면 "사장대우", "잘 대우하다"와 같은 뜻의 "대우"와 "전자, 후자"와 같은 경우로 사용되는 "전자"로 나눠진다. 그러나 "대우전자"는 하나의 고유명사로서 다른 의미구조를 가진다. 이렇게 무조건 복합명사를 단위명사로 분리하게 되면 클러스터링을 위한 문서 벡터 공간을 형성할 때, 위와 같은 단어가 포함된 문서를 구별하지 못하므로 문서 고유의 특징을 상실하게 된다. 또한 본 논문에서는 품사 태거를 사용하지 않으므로 하나의 복합명사에 대한 결과가 여러 가지일 경우에 이들 단위명사를 모두 색인어로 추출하게 된다.

참고로 본 논문에서 사용되는 형태소 분석 시스템에서의 복합명사의 분석은 〈표 1〉과 같은 알고리즘으로 수행된다(조현양, 최성필 2001).

이 분석 알고리즘의 가장 큰 특징은 입력 단어에 대해서 복합명사의 요소명사일 가능성이 있는 후보 단위명사를 추출하여 이들을 하나의

〈표 1〉 복합명사 분석 알고리즘

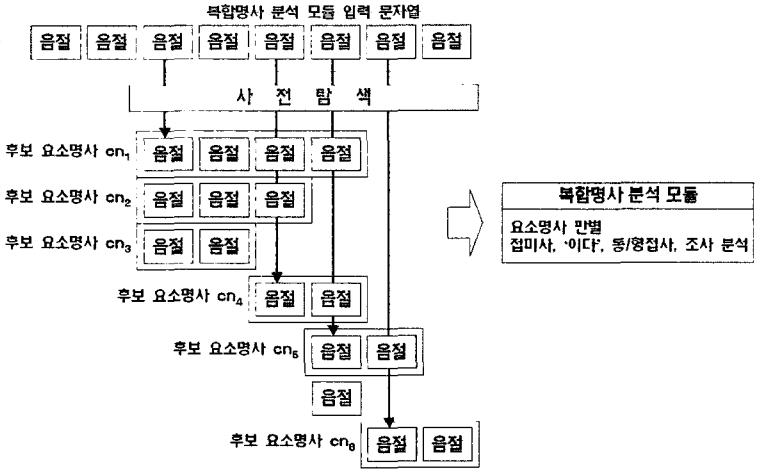
```
Ssub : 입력 어절에서의 현재 분석 대상 부분 어절
Ui : 분석 대상 음절
Ssub = Ui Ui+1 Ui+2 ... Ui+k
for (j = 0 to k-1) {
    if (j == 1) continue;
    Ui+j 위치에서 사전 탐색 수행;
    if (2음절 명사)
      { VERTEX[Index]에 현재 사전 탐색 결과 저장; }
}
연속된 명사 리스트를 찾아서 DADJ에 인덱스를 저장;
/* 후보 단위명사 그래프 traversal */
while (VERTEX를 다 검사할 때까지) {
    PUSHSTACK <-- Initial Index;
    while (!StackEmpty) {
          POPSTACK;
          현재 명사 노드 정보를 형태소 분석 결과 버퍼에 저장;
          현재 노드와 연관되는 명사 리스트 정보를 DADJ
              에서 추출하여 스택에 저장;
          접미사 검사; "이다" 조사 검사;
          용언화 접사 검사; 조사 검사;
    }
}
```

그래프로 구성한다는 것이다. 각 그래프의 입력 연결선(edge)와 출력 연결선은 각각 현재 후보 단위명사의 이전 후보 단위명사, 이후 후보 단위명사의 정점(vertex)를 가리킨다. 실제 복합명사 분석은 이 그래프가 구성된 후에 수행된다.

우선, 남은 분석 어절의 모든 위치에서 사전 탐색을 수행하여 2음절 이상의 명사를 추출하고, 추출된 2음절 이상의 명사에 대한 현재 분석 어절에서의 위치와 함께 사전 정보를 하나의 정점으로 구성한다. 이렇게 구성된 정점에 각각의 명사 위치와 길이를 검사하여 연결된 명사 리스트를 생성하고, 이를 매트릭스에 저장한다. 실제 복합명사 분석 모듈에서는 스택을 이용하여 유효한 명사 리스트를 따라가며 분석을 수행한 후, 복합 명사 분석 결과를 생성하게 된다.

다음의 〈그림 2〉는 위의 알고리즘을 이용하여 복합명사가 분석되는 개념과 구조를 나타낸다.

〈그림 2〉에서 입력 문자열의 두 번째 음절 위치를 제외한 모든 위치에서의 사전 탐색 결과로 여러 개의 후보 요소명사가 추출된다. 이 후보 요소명사를 그래프의 한 정점으로 간주하고 그래프의 각 노드를 스택을 이용하여 방문하면서 복합명사 분석 모듈을 호출하게 된다.



〈그림 2〉 복합명사 분석

만일 입력 문자열의 끝까지 분석이 성공되면 하나의 복합명사 분석 결과로 분석결과 버퍼에 저장하게 된다.

본 논문에서 수행된 복합 명사 분석 알고리즘은 모든 분석 대상 음절 위치에서의 사전 탐색에 따른 사전 탐색 횟수의 증가에 대한 문제점이 있을 수 있으나, 일반적인 복합명사 분석 알고리즘에서 사용하고 있는 재귀적 호출이나 복잡한 모듈의 구현을 피할 수 있는 장점이 있다. 만일 분석 상의 오류나 추가로 수행되어야 할 기능적 모듈을 전체 복합 명사 모듈에 추가시킬 때는 후보 요소명사 그래프 생성 부분은 수정할 필요 없이 그래프의 각 노드들을 방문하며 실제 복합명사를 분석하는 부분만을 수정하면 된다.

복합명사 분석 방법에 따른 효과를 살펴보기 위해서 본 논문에서 제시한 복합명사 분석 방법은 크게 3가지이다. 이들 분석 방법에 대한 설명은 〈표 1〉에 나타난 바와 같다.

〈표 2〉에서 제시된 CAN 방법은 전통적으로 수행되어 오던 복합명사 색인 방법으로서 복합명사를 구성하는 단위명사만을 색인어로 등재시키는 방법이다. 또한 CAOI는 이와는 반대로 단위명사는 모두 제외시키고 복합명사 자체만을 색인어로 등재시키는 방법이다. 위의 두 가지 방법을 혼합하여 CAI 방법에서는 복합명사를 단위명사로 분리하여 색인어로 등재시킨 뒤에 복합명사 자체도 색인어로 추가시키

〈표 2〉 복합명사 색인 방법

CN=(N_1, N_2, ... N_n) : 분석결과 나온 모든 단위명사들 집합		
종 류	추출 색인어	설 명
CAI	N_1,N_2,...N_n,CN	복합명사 자체를 색인어에 포함
CAN	N_1,N_2,...N_n	단위명사만을 색인어에 포함
CAOI	CN	복합명사 자체만을 색인어에 포함

는 방법이다.

3. HAC 기반 문서 클러스터링 엔진

일반적으로 문서를 그룹화 하는 방법은 크게 문서 분류와 문서 클러스터링의 두 가지로 나뉜다[8]. 두 가지 방법 모두 문서를 유사도 기준으로 분류한다는 점에서는 공통점이 있으나, 문서 분류는 분류시스템을 특정 문서 집합에 최적화시키기 위해 분류 기준(classification criterion)과 학습 문서 집합(training set)이 필요하며 이를 바탕으로 학습(learning)이 수행되어야 한다. 따라서 성능 수행 차원에서는 장점이 있으나 학습 자료를 준비해야 하고 이를 특정 문서집합에만 적용할 수가 있어서 응용성과 편리성 측면에서는 단점으로 작용될 수 있다. 이에 반하여 문서 클러스터링은 학습 문서 집합과 분류 기준이 필요 없으며, 전처리 단계인 학습단계도 불필요하다. 대신 문서에 대한 분류결과가 평면적인 그룹 형태로 나타나며, 성능이 다소 떨어질 수 있다는 단점이 있다.

문서 클러스터링 방법에는 크게 계층적 클러스터링과 비계층적 클러스터링 방법이 존재한다[8]. 계층적 클러스터링은 문서의 계층적 관계에 따라 군집화를 시키는 방법으로써 또다시 하향식(top-down) 방법과 상향식(bottom-up) 방법으로 나눌 수 있다. 하향식 방법은 전체 문서집합을 하나의 클러스터로 보고 상이한 문서 혹은 클러스터를 분리하는 과정을 수행한다. 그리고 상향식 방법은 결합 클러스터링(agglomerative clustering)이라고도 하는데 문서집합의 모든 문서 각각을 하나의 클러스터라고 보고 유사한 클러스터를 결합함으로써 군집화를 수행하는 방법이다. 비계층적 클러스터링은 상기에 언급된 문서분류(Document Classification) 기법이 포함된다. 여기에는 K-means 알고리즘, EM-Algorithm(Manning, Christopher D. and Hinrich Schutze 1999) 등이 있다. 본 논문에서는 계층적 클러스터링의 한 종류인 계층적 결합 클러스터링을 활용하였다. 또한 시스템의 복잡성을 피하고 다양한 요인에 따른 다양한 결과를 분석하기 위해 복수계층이 아닌 단일 평면계층으로만 클러스터링을 수행하도록 하였다.

본 논문에서 개발된 문서 클러스터링 엔진의 구조는 〈그림 3〉과 같다. 〈그림 3〉에서 보는 바와 같이 문서 클러스터링 시스템은 크게 2개의 모듈로 구성되어 있으며, 클러스터링 작업은 2단계로 이루어진다. 첫 번째 단계에서는 입력 문서를 색인하고 색인어를 추출하여 데이터베이스에 각종 통계정보를 저장하고 문서 벡터를 저장하게 된다. 두 번째 단계에서는 생성된 문서 벡터 정보를 이용하여 문서간 유사도와 클러스터간 유사도를 중심으로 클러스터링을 수행하게 된다. 문서 벡터를 구성하는 각 색인어에 대한 가중치 계산은 일반적으로 많이 사용되는 로그 tf*idf를 사용하였다.

$$w_{d,t} = (K + (1-K)\frac{f_{d,t}}{\max_i f_{d,i}}) \times \log\frac{N}{f_t} \qquad (1)$$

여기서 $f_{d,t}$는 문서 d내에서 단어 t의 빈도이고 f_t는 단어 t의 전체 단어 빈도이다. K값은 0으로 지정하였다. 문서간의 유사도는 코사인

계수를 그대로 적용하였다.

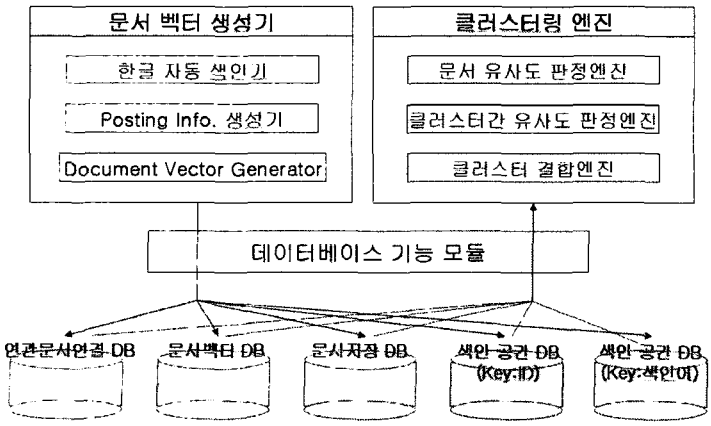
$$Sim(D_i, D_j) = \frac{D_i \cdot D_j}{|D_i||D_j|} = \frac{1}{W_i W_j}\sum_{t=1}^{n} w_{i,t} \cdot w_{j,t}$$
$$W_i = \left(\sum_{t=1}^{n} w^2_{i,t}\right)^{1/2}, W_j = \left(\sum_{t=1}^{n} w^2_{j,t}\right)^{1/2} \quad (2)$$

클러스터간 유사도 측정은 군집 평균 연결(group average link)방법을 이용하였다. 또한 문서 간 혹은 클러스터간 유사도가 특정 임계치보다 큰 두 문서 혹은 클러스터가 하나의 클러스터로 결합된다. 이런 병합은 클러스터 집합 내의 모든 클러스터쌍이 임계치보다 낮을 때까지 반복된다.

$$SimC(C_i, C_j) = \frac{1}{|C_i||C_j|}\sum_{d_i \in C_i}\sum_{d_j \in C_j} SimD(d_i, d_j) \quad (3)$$

〈그림 3〉에서, 문서 벡터 생성기는 내재된 한글 자동 색인 시스템을 이용하여 입력 문서를 색인하고 이를 바탕으로 문서 벡터 데이터베이스를 구성하는 역할을 수행한다. 각 문서당 벡터에는 그 문서에 존재하는 색인어에 대한 식별자 리스트가 저장되며, 이 색인어 식별자를 키 값으로 활용하여 실제 색인어 정보를 색인 공간 DB에서 검색할 수 있다. 최종적으로 구성된 문서 벡터 DB를 활용하여 클러스터링 엔진 모듈에서는 각 문서 벡터를 각각 비교하면서 임계치 이상의 유사도를 나타내는 문서를 하나의 클러스터로 결합하게 된다.

클러스터링 과정에서 문서 집합 내의 모든 문서에 대해서 유사도를 측정해야하는 부담을 덜기 위해서 문서 벡터 생성 시에 같은 색인어를 공유하는 문서를 선택하여 연관 문서 연결 DB에 저장하게 된다. 이렇게 하면 클러스터링 과정에서 특정 문서에 대한 연관 문서에 대해서만 유사도 측정을 하면 되므로 유사도가 특정 임계치 이하의 문서 쌍은 유사도 측정을 하지 않게 된다. 문서 벡터 생성 단계는 문서 클러스터링 결과와는 직접적인 관련이 없으므로 문서 벡터 DB를 생성한 다음 임계치를 조정해 가면서 상황에 맞는 문서 클러스터링을 수행할 수 있다.



〈그림 3〉 문서 클러스터링 엔진의 구조

4. 실험 및 결과 분석

본 논문에서 개발된 자동 색인 시스템과 문서 클러스터링 엔진에 대한 실험은 한국일보에서 제공한 3개월 분의 신문기사를 이용하여 수행되었다. 기사의 내용은 정치, 경제, 사회 등 다양한 내용을 담고 있으며, 데이터에 대한 임의적인 조작 및 전처리 과정은 거치지 않았다. 본 논문에서 사용된 실험 데이터에 대한 설명은 다음 〈표 3〉과 같다.

실험은 〈표 2〉에서 제시한 3가지 복합명사 색인 방법에 따라서 진행되었다. 각 유형별로 불용어 제거 기준임계치와 유사도 판정임계치에 따른 클러스터링 결과의 변화를 관찰하였다. 우선 전체 데이터에 대해서 색인 작업을 수행하여 나온 유형별 색인어의 개수와 전체 색인어 개수와 제거된 불용어의 개수의 비율을 아래 〈표 4〉와 〈그림 4〉에 나타내었다.

단일화된 색인어 집합은 전체 실험문서를 자동 색인하여 구해진 색인어 집합을 분석하고, 중복을 제거하여 전체 색인어 개수를 조사한 것이다. 중복되지 않은 전체 색인어 개수와의 비율을 보면 CAI는 약 5%, CAN은 약 3%, CAOI는 약 7% 정도에 그친다. 따라서 중복으로 활용되는 색인어가 전체 색인어의 약 93%에서 97%까지에 이름을 알 수 있다. 그리고 CAN 방법에서 단일화된 색인어 수가 감소함을 알 수 있다. 이는 다수의 복합명사에 중복으로 포함된 단위명사들이 복합명사가 분리되면서 단일명사로 인식되어 중복 제거되었기 때문이다. 예를 들어, "학교생활"과 "생활지도"라는 복합명사에 중복으로 포함된 "생활"이 하나로 단일화되면서 전체 색인어 개수가 줄어드는 현상을 나타내고 있다.

〈표 5〉는 〈표 2〉에서 제시한 복합명사 분리 색인방법이 어떻게 이루어지는 가에 대한 예시를 보여준다. CAI는 단위명사와 함께 복합명사 자체까지도 색인어로 포함시키고 있고, CAOI는 복합명사 자체만을 색인어로 포함시키는 방법이다.

〈그림 4〉는 옵션으로 지정할 수 있는 제거(cut-off) 임계치 별로 각 복합명사 분리 방법을 적용함에 있어서 제거된 색인어 개수를 그래프로 나타낸 것이다. 여기서 cut-off 값은

〈표 3〉 실험 데이터 정보

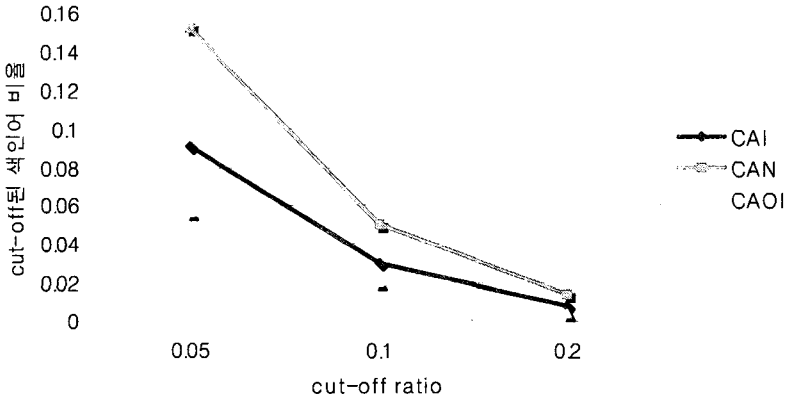
데이터	한국일보기사(1999년 9월 - 1999년 11월)
건 수	23,450(건)
크 기	39.309(Mbyte)

〈표 4〉 실험 데이터 분석 정보

항 목	유 형		
	CAI	CAN	CAOI
단일화된 색인어 개수	297,731	176,651	286,966
전체 색인어 개수	5,260,083	4,602,929	3,995,321
색인어당 평균 출현빈도	17.667	26.057	13.923

〈표 5〉 복합명사 색인 방법의 예

CN=(N_1,N_2,...N_n) : 분석결과 나온 모든 단위명사들 집합		
종 류	추출 색인어	설 명
CAI	N_1,N_2,...N_n,CN	"정보"+"검색"+"정보검색"
CAN	N_1,N_2,...N_n	"정보"+"검색"
CAOI	CN	"정보검색"



〈그림 4〉 색인어 Cut-Off 비율

색인어 가중치(식(1)) 값으로써 임계치(threshold)가 특정 값 이하이면 그 색인어는 불필요한 색인어로 인식되어 제거된다. 〈그림 4〉에서 Y축은 제거된 불용어의 개수를 각 유형별 단일화된 색인어 개수로 나누어서 백분율로 나타낸 값이다. 위 그림에서 보는 바와 같이 CAN 타입으로 색인을 수행하였을 때 제거율이 가장 높게 나타난다. 이는 복합명사에서 분리된 단위명사가 일반적인 성향을 다수 띄어서 DF가 높게 나타나는 명사들이 많음을 보여준다. CAOI 타입에서 제거율이 낮은 이유는 복합명사가 분리되지 않으므로 복합명사를 구성하는 단위명사들이 제거되지 않기 때문이다.

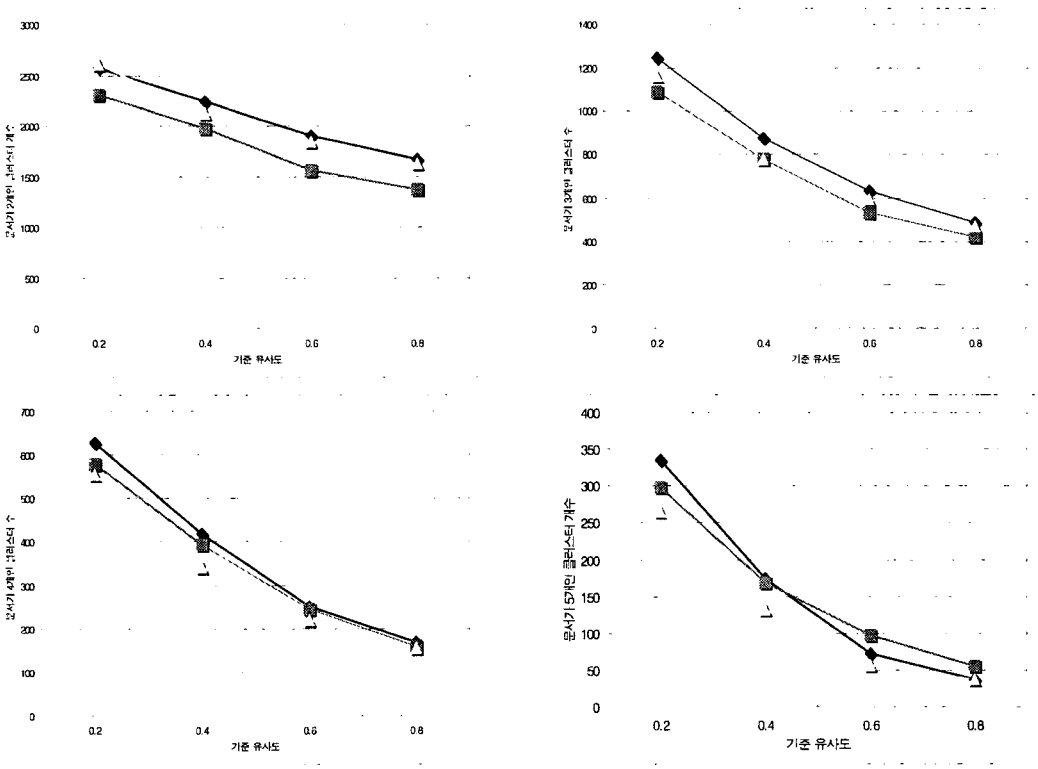
〈표 6〉은 유사도 임계치별 클러스터 개수를 보여준다. 앞에서도 지적했듯이 기준 유사도는 두 개의 클러스터가 하나의 클러스터로 병합되는 기준 값이다. 예를 들어, 기준 유사도가 0.4일 때, 두 클러스터의 유사도가 0.4보다 크면 하나의 클러스터로 합쳐지게 되는 것이다. 평균적으로 CAI가 가장 많은 클러스터를 생성하고 CAN이 가장 적은 클러스터를 생성하게 된다. 모든 색인방법 공통으로 기준 유사도의 변화에 따른 결과 클러스터의 개수변화의 추이를 살펴보면, 기준 유사도가 0.2에서 0.4로 바뀔 때 클러스터 수가 가장 많이 줄어들었으며, 유사도가 점차 높아짐에 따라서 그 차이는 점점 줄어들게 된다. 여기서 주의 깊게 살펴봐야 할 점은 외형적으로 볼 때 CAI와 CAN의 색

〈표 6〉 유사도 임계치별 클러스터 개수

기준 유사도	CAI	CAN	CAOI
0.2	5,118	4,601	4,842
0.4	3,832	3,452	3,477
0.6	2,900	2,484	2,752
0.8	2,390	2,018	2,319
평균	3,560.00	3,138.75	3,347.50

인 방법이 유사하게 느껴질 수가 있으나 오히려 결과 클러스터의 개수는 CAI와 CAOI가 더 유사하게 나타난다는 점이다. 이는 복합명사를 구성하는 단위명사보다는 복합명사 자체가 클러스터링 결과에 많은 영향을 미친다는 것을 반증하는 것으로 보인다. 다시 말해서, "대우전자"라는 복합명사가 분리된 형태인 "대우"와 "전자"만으로 구성된 문서벡터가 "대우전자"라는 색인어로만 구성된 문서벡터보다 변별력이 떨어진다는 것이다.

〈그림 5〉는 각 유형별 기준 유사도에 따른 포함 문서가 2,3,4,5개인 클러스터 개수 변화 추이를 나타낸다. 그림에서 마름모선은 CAI를 나타내며, 네모선은 CAN, 그리고 삼각형은



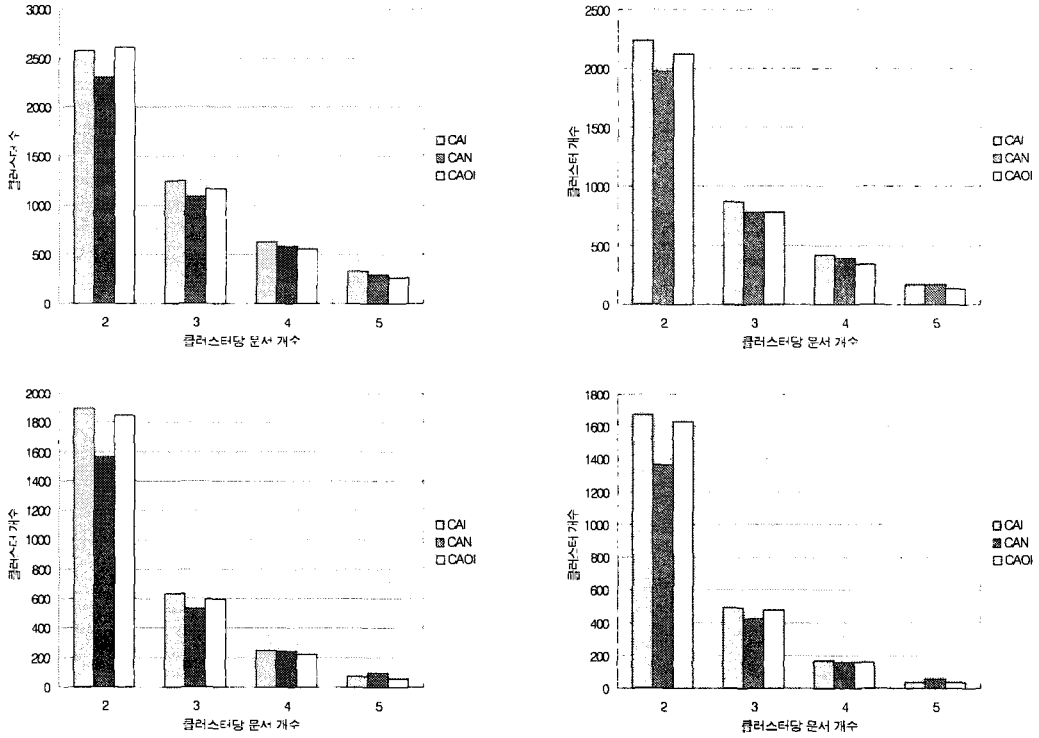
〈그림 5〉 문서개수가 2,3,4,5인 클러스터 개수 변화 추이

CAOI의 클러스터링 결과를 나타낸다.

위 그래프에서, CAOI는 포함문서가 2개 혹은 3개인 클러스터의 개수가 CAN보다 더 많은 반면에 포함문서가 4개 혹은 5개인 클러스터의 개수는 CAN보다 적음을 알 수 있다. 특히 포함문서가 2개인 클러스터의 개수는 CAI와 거의 흡사함을 보여준다. 또한 CAN은 포함문서가 2개 혹은 3개인 클러스터의 개수가 비교적 적은 반면, 포함문서가 4개 혹은 5개인 클러스터의 개수는 오히려 다른 방법에 비해서 증가하고 있음을 알 수 있다. 각 유형별로 포함문서가 2개인 클러스터의 개수는 기준 유사도의 변화에 따른 값의 변화가 적다. 그러나 3개 이상인 클러스터의 개수는 기준 유사도의 변화에 따른 클러스터 개수의 변화가 큰 것을 알 수 있다. 더불어, 기준 유사도가 0.8일 경우에는 모든 방법에서 거의 동일한 결과로 수렴하고 있음을 알 수 있다.

따라서 복합명사에 대한 색인방법은 최종적으로 도출되는 결과 클러스터의 형태 및 개수에 많은 영향을 미치고 있으며 기준 유사도와의 관계에 있어서도 많은 상관관계가 있음을 알 수 있다.

〈그림 6〉은 〈그림 5〉와는 다른 관점에서 분석한 결과로서 기준 유사도별로 클러스터 당 문서 개수를 기준으로 하여 클러스터의 개수를 각 유형별로 분석한 그래프이다. 위쪽 그래프의 왼쪽은 기준 유사도 0.2, 오른쪽은 0.4, 아



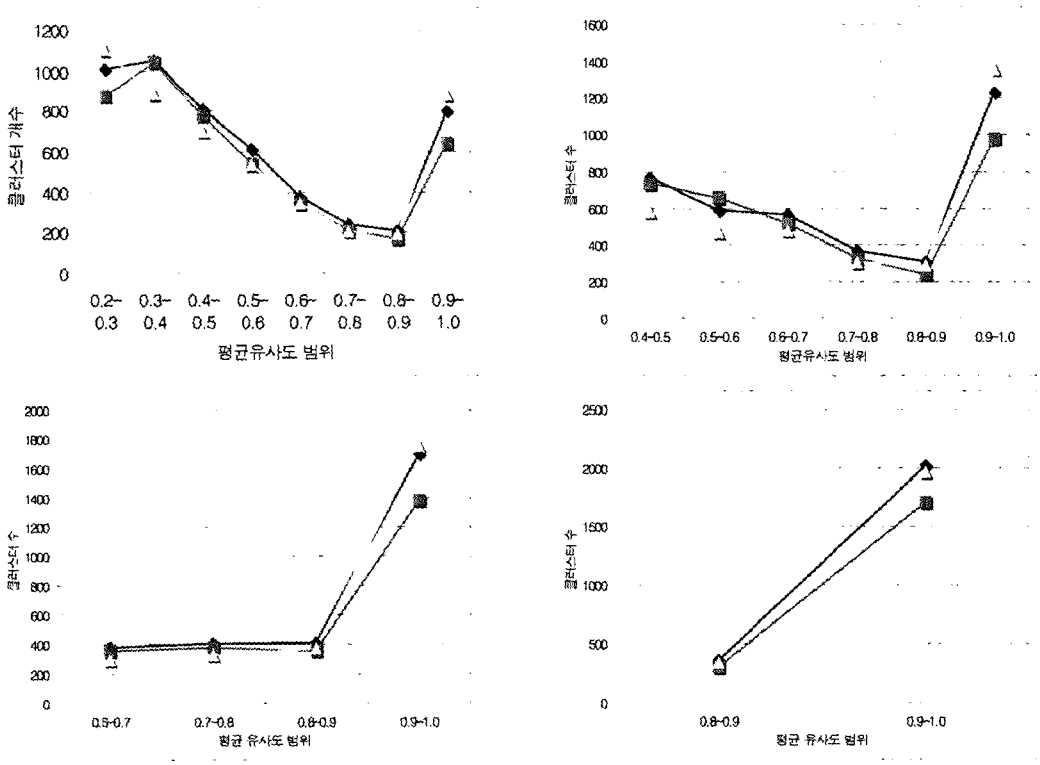
〈그림 6〉 기준 유사도별 클러스터당 문서개수 기준 클러스터 수

래 그래프의 왼쪽이 0.6, 오른쪽이 0.8일 때의 클러스터 결과이다.

매 기준 유사도에 따른 포함 문서가 2개인 클러스터의 개수 변화는 각 유형별로 변동이 적다. 그러나 포함 문서가 3개 이상인 클러스터의 개수 변화는 비교적 많이 나타나고 있음을 알 수 있다. 포함 문서 개수가 2개인 클러스터는 CAI와 CAOI 유형에서 많이 나타나지만 4개 이상인 클러스터는 CAN에서 비교적 많이 나타남을 보여준다.

〈그림 7〉은 기준 유사도의 변화추이에 따른 클러스터 내의 문서들 간의 평균 유사도 범위에 따른 클러스터의 개수를 나타낸다. 상기에 언급된 바와 같이 〈그림 7〉에서 마름모선은 CAI를 나타내며, 네모선은 CAN, 그리고 삼각형은 CAOI의 클러스터링 결과를 나타낸다. 위쪽 그래프의 왼쪽은 기준 유사도 0.2, 오른쪽은 0.4, 아래 그래프의 왼쪽이 0.6, 오른쪽이 0.8일 때의 클러스터 결과이다. 각 그래프의 평균유사도 범위 최소 값은 기준 유사도 값이 된다.

평균 유사도는 한 클러스터 내의 모든 문서 쌍에 대한 유사도 평균값이다. 이 값은 특정 클러스터의 응집도를 나타내기도 한다(강동혁, 주길홍, 이원석 2001). 평균 유사도 범위가 0.2-0.5인 곳을 제외하고는 세 가지 유형의 값 변화가 같은 형태로 나타난다. 또한 만일 기준 유사도와 클러스터 내의 평균 유사도 사이



〈그림 7〉 평균 유사도 범위 기준 클러스터 개수 변화 추이

의 관계가 전혀 없다면, 위 네 개의 그래프는 동일한 형태를 나타내야 하나, 조금씩 차이를 나타내고 있는 것을 보면 HAC 알고리즘과 기준 유사도, 그리고 이를 통해 결과로 나오는 클러스터 집합의 평균 유사도 사이에는 어느 정도 연관성이 있음을 알 수 있다. 위 그림에서 CAOI 유형은 조금 특이한 형태를 취하고 있다. 평균 유사도 범위가 0.2-0.3인 경우를 제외하고는 0.4-0.5, 0.6-0.7 등의 범위 즉, 기준 유사도에 근접한 평균 유사도를 가지는 클러스터의 개수는 세 유형 가운데 가장 낮지만, 가장 높은 평균 유사도인 0.9-1.0 사이의 클러스터는 가장 많이 나타난다. 또한 세 유형 모두 평균 유사도 0.8-0.9에서 클러스터의 수가 거의 동일하게 나타나고 있다.

5. 결 론

본 논문에서는 구조화된 형태소 분석기를 이용하여 복합명사의 분석 유형을 다양하게 적용한 계층적 결합 문서 클러스터링 엔진을 개발하였다. 그리고 이를 통하여 결과로 나오는 클러스터 집합에 대한 실험 및 분석을 수행하였다. 클러스터링 결과에 대한 정확도는 테스트 집합의 부재로 실험하지 못하였으나, 실험 데이터를 이용한 테스트 결과로는 CAI 유형의 클러스터링 집합이 사용자 만족도 면에서 가장 높은 점수를 획득하였다. 객관적인 평가자 선정 및 체계적인 만족도 평가가 이루어지지 않은 관계로 평가 결과를 수치로 제시할 수는 없으나, 이는 일반적인 단위명사 분리의 이점과 복합명사의 단일의미를 나타내는 복합명사 자체 색인어의 이점이 부가되어서 나온 결과로 보인다.

본문에서는 언급하지 않았으나 본 논문에서 사용된 실험 데이터를 분석한 결과 약 30% 정도의 어휘가 복합명사로 분석되고 있었다. 전체 데이터 중 30%를 차지하는 복합명사에 대한 색인방법 변형이 가져온 클러스터링 결과의 변화는 많은 시사점을 준다. 우선 외형적으로 검토해 보았을 때, 복합명사를 구성하는 단위명사에 복합명사 자체도 함께 포함시켜 색인하는 CAI 방법과 단위명사만을 색인어에 포함시키는 CAN, 이 두 방법이 더 유사하게 보일 수 있다. 그러나 본 실험 결과에서 나타난 바와 같이 CAI 방법은 오히려 복합명사 자체만을 색인어로 등재시키는 CAOI 방법과 훨씬 더 유사한 결과를 나타내었다. 이는 복합명사가 특정 문서에 대한 변별력을 보다 많이 포함하고 있음을 나타내는 것이며, 이 결과를 정보검색시스템이나 다른 지능형 시스템에 적용시킬 수 있는 단초를 제공할 수 있다.

본 논문에서 다양한 형태의 실험결과 분석을 통해 복합명사의 분석방법에 따른 클러스터링 결과에 대한 연구를 수행하였으나, 보다 체계적이고 이론적인 연구가 복합명사 분석방법과 클러스터링 알고리즘의 연관관계에 대한 분석에 진행되어야 한다. 무엇보다도 문서 클러스터링 엔진에 대한 정확한 성능 측정을 위해서 문서 클러스터링 테스트 집합이 마련되어야 한다. 본 논문에서 수행된 실험 결과에 대한 정확하고 정형적인 모델을 확립하고 이를 통한 보다 세부적인 연구도 향후에는 필요할 것이다.

참 고 문 헌

강동혁, 주길홍, 이원석. 2001. 의미정보의 효율적인 분류를 위한 계층적 중복 문서 클러스터링. 『한국정보과학회 2001 가을 학술발표논문집』.

강승식. 1993. 『음절 정보와 복수어 단위정보를 이용한 한국어 형태소 분석』. 서울대학교 컴퓨터공학과 박사학위논문.

심철민. 1995. 『어절간 연관관계와 오류 유형 추정 규칙에 기반한 한국어 철자교정기』. 부산대학교 전자계산학과 석사학위논문.

조현양, 최성필. 2001. 어절 분석 기반 형태소 분석 시스템 개발에 관한 연구. 『한국정보관리학회논문지』, 018(002): 105-124.

최성필, 서정현, 채영숙. 2002. 자동 색인을 위한 한국어 형태소 분석기의 실제적인 구현 및 적용. 『한국정보처리학회논문지』.

최성필. 1998. 『오류분석정보와 복합명사의 의미처리규칙 및 말뭉치를 이용한 철자교정기의 성능개선』. 부산대학교 전자계산학과 석사학위논문.

채영숙, 김재원, 김민정, 권혁철. 1991. 한국어 철자 검색을 위한 형태소 분석 기법, '91 우리말 정보화 잔치. 『국어정보학회』, 179-186.

Baeza-Yates, Ricardo, and Ribeiro-Neto, Berthier, 1999, *Mordern Information Retrieval*, New York: ACM Press.

Charniak, Eugene, 1993, "Statistical Language Learning", A Bradford Book, Cambridge: The MIT Press.

Ian H. Witten, Alistair Moffat, Timothy C. Bell, 1994, "Managing Gigabytes", Van Nostrand Reinhold.

Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski, 2000, "DATA MINING Methods for Knowledge Discovery", Kluwer Academic Publishers.

Manning, Christopher D. and Hinrich Schutze, 1999, "Foundations of Statistical Natural Language Processing", Cambridge: The MIT Press.

Sheldon Ross, 2002, "A First Course in Probability", Prentice Hall.