대용량 문서 데이터베이스를 위한 효율적인 점진적 문서 클러스터링 기법

강 동 혁[†] · 주 길 홍[††] · 이 원 석[†††]

요 약

컴퓨터의 발전과 인터넷의 급속한 발전으로 정보의 양이 폭발적으로 증가하게 되었고 이러한 방대한 양의 정보들은 대부분 문서 형태로 관리되고 있으며, 문서 단위로 표현된 많은 정보들을 효과적으로 관리하고 검색하기 위한 방법의 연구가 필요하게 되었다. 문서 클러스터링은 문서간의 유사도를 바탕으로 서로 연관된 문서들을 군집화하여 문서들을 주제별로 통합하는 방법으로 대용량의 문서들을 자동으로 분류하고, 검색하는 데 있어서 검색의 정확성을 증대시킬 수 있다. 본 논문에서는 새로운 문서의 추가나 기존문서의 삭제로 인하여 군집화 대상이 되는 문서 집합이 점진적으로 변화하는 환경을 위한 점진적 문서 클러스터링 알고리즘을 제안한다. 점진적 문서 클러스터링 알고리즘은 새로운 문서가 추가되었을 경우 문서 전체를 다시 클러스터링하지 않고, 이미 생성된 클러스터들의 구조를 적응적으로 변화시킴으로써 높은 효율성을 제공할 수 있다. 또한, 문서 클러스터링의 정확도를 높이기 위하여 통계적인 기법으로 불용어를 판별하여 제거하는 알고리즘을 제안하고, 문서 클러스터링에서 정확한 단어가중치 산출을 위해 TF×IDF 공식을 수정한 TF×NIDF 공식을 제안한다.

An Effective Incremental Text Clustering Method for the Large Document Database

Dong Hyuk Kang[†] · Kil Hong Joo[††] · Won Suk Lee[†††]

ABSTRACT

With the development of the internet and computer, the amount of information through the internet is increasing rapidly and it is managed in document form. For this reason, the research into the method to manage for a large amount of document in an effective way is necessary. The document clustering is integrated documents to subject by classifying a set of documents through their similarity among them. Accordingly, the document clustering can be used in exploring and searching a document and it can increased accuracy of search. This paper proposes an efficient incremental clustering method for a set of documents increase gradually. The incremental document clustering algorithm assigns a set of new documents to the legacy clusters which have been identified in advance. In addition, to improve the correctness of the clustering, removing the stop words can be proposed and the weight of the word can be calculated by the proposed TF×NIDF function.

키워드 : 문서 클러스터링 기법(Document Clustering Method), 점진적 클러스터링(Incremental Clustering), 불용어 추출(Stop Word Extraction)

1. 서 론

현대 사회는 정보화 사회라 불릴 만큼 정보의 양이 기하급수적으로 증가하고 있다. 특히, WWW(World Wide Web)의 발전으로 인터넷을 통해 접하게 되는 정보의 양은 빠른 속도로 증가하고 있다. 최근에는 웹 문서에 이미지, 사운드, 동영상 등 멀티미디어 정보가 많이 포함되고 있지만 여전히 정보의 주된 전달 형태는 텍스트 형태의 정보이다. 현재 인터넷 상에 존재하는 텍스트 정보의 양은 수백 기가 바이트에 달하며, 앞으로 더욱 증가할 것이다. 정보의 급격한 증가로 인해 인터넷 검색 엔진과 같은 정보 검색 시스템에서 정보를 효율적으로 검색하기 위해 많은 연구가 진행되고 있다. 현재 문서 검색 방법에서 많이 사용되고 있는 방법은 미리 정해진 분류 카테고리에 따라 사용자가 검색하고자 하는 문서를 탐색해 나가는 방법과 사용자가 검색어를 입력하면 데이터베이스에 저장되어 있는 색인이나 문서에 포함된 단어와 일치하는 문서를 찾아서 사용자에게 제시하는 방법이다[17]. 이 중에서도 문서 분류(Document Classification)나 문서 클러스터링(Document Clustering)과 같은 방법은 정보 검색 시스템에서 방대한 양의 문서들을 구조화하는데 중요한 역할을 담당하고 있다. 예를 들어, 인터넷 검색 엔진에서는 웹 사이트나 웹 문서들을 사전에 분류하여 사용자가 검색하고자 하는 영역을 지정할 수 있다.

† 정 회 원 : (주)네트빌 부설연구소 연구원
†† 준 회 원 : 연세대학교 대학원 컴퓨터과학과
††† 종신회원 : 연세대학교 컴퓨터과학과 교수
논문접수 : 2002년 10월 11일, 심사완료 : 2002년 11월 5일

그 뿐만 아니라, 문서 분류를 이용하여 사용자가 입력한 질의어에 대한 검색 결과를 보다 효율적으로 탐색할 수 있는 수단을 제공한다[9, 10]. 이러한 이유 때문에 문서 분류나 클러스터링 방법이 차세대 검색 방법의 중심기술로 등장하고 있다. 지금까지 문서 분류는 주로 숙련된 전문가에 의해 작성되고 있다. 그러나, 새로운 문서와 새로운 웹사이트들이 계속해서 생겨나기 때문에 시간이 지날수록 사람에 의한 문서 분류는 한계를 가질 수밖에 없으므로 방대한 양의 문서들을 정확하게 구조화하여 문서 검색의 효율성을 높일 수 있는 자동화된 문서 분류 방법에 대한 연구가 필요하다.

본 논문에서는 자동화 문서 구조화의 정확성을 높이기 위해 불필요한 단어를 삭제하는 불용어 제거 알고리즘을 제안한다. 불용어 제거 알고리즘을 통해 각 문서의 불용어를 제거한 후에 해당 문서의 주제어를 추출하여 이를 기반으로 클러스터링 과정을 수행한다. 문서 클러스터링 방법을 상대적으로 적은 수의 초기 문서 집합에 적용하여 초기 문서 클러스터들을 찾고 이를 기반으로 초기 문서 카테고리를 자동으로 생성한다. 그 후 새로 추가되는 문서에 대해 문서의 주제어 기반으로 이전 카테고리를 이용하여 해당되는 클러스터를 찾고 클러스터의 최소 응집도와 최소 참여도를 기반으로 점진적으로 해당 클러스터에 대한 새로운 문서의 추가여부를 결정하여 문서의 추가가 가능한 경우에 해당 클러스터의 문서와 새문서를 재클러스터링하여 보다 명확하게 군집화 할 수 있는 클러스터들을 찾는 점진적 문서 클러스터링 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 기술하고, 3장에서는 계층적 문서 클러스터링 방법을 위한 불용어 제거 알고리즘과 주제어 선정방법 및 이를 이용한 초기 계층적 문서 클러스터링 방법에 대해서 기술한다. 4장에서는 문서 카테고리 트리를 이용한 점진적 클러스터링 방법에 대해서 기술하고, 5장에서는 3장과 4장에서 제안한 내용에 대한 다양한 실험을 수행하고 이에 대한 결과를 분석한다. 마지막으로 6장에서는 최종적인 결론을 맺는다.

2. 관련 연구

많은 문서에 대한 효율적인 검색을 지원하기 위해 시도되고 있는 방법으로 문서 분류와 문서 클러스터링이 있다. 먼저 문서 분류는 특정 문서가 미리 정의된 문서 집합 중에서 어느 집합에 속하게 되는지를 결정하는 작업이다[8]. 지금까지의 문서 분류 작업은 숙련된 전문가에 의해서 직접 수행되어졌다. 예를 들어, '야후'[5]에서는 현재 숙련된 문서 분류 전문가에 의해 새로운 웹 사이트를 현존하는 카테고리에 배정하는 방법을 사용하고 있다. 전문가의 수작업의 의한 방식들의 한계를 극복하기 위해 문서 분류를 자동화할 수 있는 효과적인 방법들의 연구가 활발하게 진행되고 있다[9, 10]. 문서 분류를 위한 자동화 방법 중 k-NN(k-nearest neighbor) 분류 방법은, 새로운 문서와 유사도를 비교하여 가장 유사한 k개의 문서들의 카테고리를 결합함으로써 새로운 문서를 분류하는 통계적인 알고리즘이다[11]. 이 방법은 특정 문서와 이웃 문서들의 유사도를 정확하게 계산하나 문서간의 유사도 계산을 위하여 문서의 모든 특성을 사용한다[10]. 문서의 자동분류를 위해 기계 학습(Machine Learning) 기술을 이용하여 문서 분류자(Text Classifier)를 학습시키는 방법들[9]은 적절히 준비된 훈련 예들을 통해서 문서 분류자를 학습시키기 때문에 분류자의 성능은 결국 훈련 예들에 크게 영향을 받는다. 따라서, 이러한 방법은 새로운 개념과 새로운 용어들이 대상문서에 있을 경우 적절하게 대처하기가 어렵다.

이와는 반대로 문서 클러스터링 방법은 문서간의 유사도를 바탕으로 연관된 문서들을 클러스터링 함으로써 문서들을 주제별로 통합하는 방법이다[1]. 문서 클러스터링 방법은 많은 수의 문서들을 분류하고, 검색하는 데 있어서 효율적으로 검색의 정확성을 증대시키는 방법으로 이에 대해 많은 연구가 활발하게 이루어지고 있다. 문서 클러스터링 방법과 관련하여 가장 널리 적용되고 있는 방법에는 반복 클러스터링(Iterative Clustering) 방법과 계층적 집적 클러스터링(HAC, Hierarchical Agglomerative Clustering) 방법이 있다[2, 6]. 반복 클러스터링 방법은 재배치(Reallocation) 방법이라고도 하며, 반복적으로 문서들을 가장 유사한 클러스터에 재할당함으로써 문서 클러스터링을 최적화하는 방법이다. 반복 클러스터링 방법에는 Buckshot과 Fractionation[1], 단일 패스(Single Pass) 알고리즘[2], 그리고 K-means 알고리즘[7]이 있다. Buckshot과 Fractionation 알고리즘은 시드 기반 분할 클러스터링 방법에서 초기의 클러스터 중심을 찾기 위한 알고리즘이다. 이 알고리즘들은 오직 초기 클러스터의 중심을 찾기 위한 알고리즘들로 다른 클러스터링 알고리즘과 함께 사용된다. 단일 패스 알고리즘은 먼저, 첫 번째 문서를 첫 번째 클러스터로 만들고, 다른 문서들을 클러스터들과 비교하여 유사도 임계치를 만족하는 클러스터 중에서 유사도가 가장 높은 클러스터에 문서를 할당하고, 모든 클러스터가 임계치를 만족하지 못할 경우 새로운 클러스터를 생성한다. 이러한 과정을 반복함으로써 문서 클러스터링을 수행한다. K-means 알고리즘은 최종적으로 생성되어야 할 클러스터의 개수를 미리 설정해 놓고 거리가 가장 멀리 떨어져 있는 문서들을 각 클러스터의 중심으로 하여 문서들에 대해 클러스터의 중심과의 거리를 비교하여 가장 가까운 클러스터에 할당하는 방법이다. 그리고, 문서들을 클러스터에 추가할 때마다 클러스터의 중심을 재조정한다.

계층적 집적 클러스터링 방법은 각 문서를 하나의 클러스터로 두고, 각 클러스터간의 유사도를 모든 클러스터 사이에 계산하여, 가장 가까운 클러스터를 새로운 클러스터로 결합하는 방법이다. 가장 유사한 두 클러스터를 결합함으로써, 클러스터의 수가 1만큼 감소하게 되고, 이를 적정수의 클러스터가 남을 때까지 반복한다. 계층적 집적 클러스터링 방법에는 단일 연결(Single Link), 완전 연결(Complete Link), 그

리고 집단 평균 연결(Group Average Link) 방법이 있다[12]. 반복적 클러스터링 방법은 수행 시간이 빠르지만, 계층적 집적 클러스터링 방법보다 클러스터링 결과의 정확도가 떨어진다[2, 6]. 따라서, 최근에는 이를 기반으로 하는 점진적 기법으로 문서 클러스터링을 수행하는 방법들이 많이 연구되고 있다[11, 12].

3. 계층적 문서 클러스터링

본 논문에서 제시하는 계층적 문서 클러스터링 방법은 다음의 세 단계로 이루어진다. 첫 번째 단계는 문서에서 명사를 추출하고 추출된 명사 중에서 불용어를 선별하여 제거함으로써 주제어 후보 단어를 선정하는 단계이다. 두 번째 단계는 첫 번째 단계에서 선택된 주제어 후보 단어별로 각각 가중치를 계산하여 주제어를 추출하는 단계이다. 세 번째 단계는 두 번째 단계에서 구해진 주제어의 가중치들을 바탕으로 계층적 집적 클러스터링 방법을 사용하여 문서 클러스터들을 찾는 단계이다.

3.1 불용어 제거

문서 내에서 단어의 빈도수는 단어의 중요도를 나타내는데 유용한 척도로 사용되지만[8] 오히려 매우 높은 빈도수를 갖는 단어는 보편적으로 널리 사용되는 일반적인 단어로써 문서의 주제어로 사용되기에는 부적합할 수 있다. 이처럼 높은 빈도수를 갖는 단어들을 불용어(Stop Word)[15]라 정의하며, 영어에는 'a, an, the, this' 등이, 한국어에는 '그러나, 나, 우리' 등이 불용어에 해당된다. 많은 검색 엔진들은 자체 불용어 목록을 가지고 있으며, 문서 색인화 과정에서 불용어들을 제거할 뿐만 아니라 사용자가 입력한 검색어에서도 불용어를 제거하고 검색을 수행하여 검색의 정확도를 향상시킨다. 이와 마찬가지로, 문서 처리 분야에서도 전처리 과정으로 불용어 목록을 이용하여 불용어들을 제거함으로써 문서 처리의 정확성을 높이고, 문서 처리 과정의 속도를 향상시킬 수 있다. 그러나, 전처리 과정에서 이용되는 불용어 목록은 다음과 같은 두 가지의 문제점이 있다. 첫째로 불용어 목록은 불용어 생성자의 주관에 따라서 불용어가 빠져있거나 불용어가 아닌 단어들이 불용어로 선택되어 있을 수 있다. 둘째로 단어의 중의성으로 인해 불용어의 판단이 모호할 수 있는 문제점이 있다. 이러한 문제점을 극복하기 위해서 본 논문에서는 해싱(Hashing) 기법[18]을 이용하여 불용어를 판별하고 제거하는 방법을 제시한다.

예를들어 불용어 제거를 위해 문서들이 정치, 경제, 사회, 과학 등과 같이 여러 대주제들로 미리 분류되어 있다고 가정하자. 이 때 분류되어 있는 문서의 집합을 *주제영역(subject domain)*라고 정의한다. 불용어 제거 알고리즘의 기본 원리는 여러 주제영역에서 많은 빈도수를 가지며 고르게 분포되어 있는 단어를 불용어로 판별하는 것이다. 본 논문에서는 불용어 판별을 위해 단어의 *주제영역 지지도(subject domain Support)*와 주제영역 내에서의 문서 지지도(Document Support)를 다음과 같이 정의한다.

[정의 1] 주제영역 지지도(Subject domain support)

주제영역 지지도란 전체 주제영역 중에서 단어가 나타나는 영역의 비율을 의미하며, 단어 t_k의 주제영역 지지도 ss_k는 식 (1)과 같이 정의된다.

$$ss_k = \frac{(\text{단어 } t_k\text{를 포함하는 영역 수})}{(\text{전체 영역 수})} \tag{1}$$

[정의 2] 문서 지지도

문서 지지도란 하나의 영역 내에서 특정 단어가 나타나는 문서의 비율을 의미하며, 영역 S_j에서 단어 t_k가 갖는 문서 지지도 ds_{jk}는 식 (2)와 같이 정의된다.

$$ds_{jk} = \frac{(\text{영역 } S_j \text{에서 단어 } t_k\text{를 포함하는 문서 수})}{(\text{영역 } S_j \text{에 포함된 전체 문서 수})} \tag{2}$$

또한, 해시 테이블은 문서 지지도를 이용하여 단어 t_k의 불용어 여부를 판별하기 위한 해시 함수로 식 (3)과 같이 나타낸다.

$$h(ds_{jk}) = \begin{cases} N_B - 1, & \text{if } ds_{jk} = 1.0 \\ \lfloor ds_{jk} \times N_B \rfloor, & otherwise \end{cases} \tag{3}$$

(단, N_B는 버킷의 수)

본 논문에서는 불용어 제거를 위해 네 가지의 매개변수를 사용한다. 먼저 버킷을 통합할 수 있는 범위를 나타내는 척도로 $\delta(0 \le \delta \le 1)$를 사용한다. 버킷과 문서 지지도의 평균의 차가 δ 안에 포함되면 통합할 수 있다고 판단한다. 또한, 불용어 제거를 위한 기준으로 최소 주제영역 지지도와 최소 문서 지지도를 사용한다. 그러나, 특정단어가 위의 불용어 제거 기준을 만족하지만 특정한 영역에서는 높은 문서지지도를 가질 수 있다. 이런 경우 특정 단어가 그 영역에서는 불용어가 아닌 주제어로 사용되는 경우이다. 따라서, 특정 문서 지지도를 사용하여 이보다 큰 값을 갖는 영역에서는 불용어로 간주하지 않는다. 따라서, 임의의 단어가 주어졌을 때 불용어 여부를 판별하여 제거하는 과정은 다음과 같다.

단계 1 : 각 영역별로 [정의 2]에 의해 임의의 단어의 문서 지지도를 구한다.

단계 2 : 문서지지도와 식 (3)의 해시함수를 통해 [주제영역 지지도, 문서 지지도 평균]를 해시 테이블의 해당 버킷에 할당한다.

단계 3 : 주제영역 지지도가 높으면서, 포함하고 있는 영역들에서의 문서 지지도의 평균이 가장 높은 버킷을 구한다.

단계 4 : 단계 3에서 구한 버킷들 중에서 문서 지지도의 평균이 δ를 만족하는 버킷들을 하나로 통합한다.

단계 5 : 통합된 버킷에 포함된 주제영역 지지도가 최소

주제영역 지지도 보다 크고, 문서 지지도의 평균이 최소 문서 지지도 보다 크다면 이 단어는 불용어 후보 단어로 간주한다.

단계 6 : 불용어 후보 단어가 특정 문서 지지도 보다 큰 값을 갖는 영역을 제외한 영역에서만 불용어로 간주하여 제거한다.

예를 들어 "학교"라는 단어가 총 5개의 영역에서 <표 1>과 같은 문서지지도를 갖는다고 하고, 총 버킷의 수가 10이라고 하자. 각 버킷에 포함되는 주제영역 지지도와 각 영역의 문서 지지도의 평균은 <표 2>와 같다. 이때 δ는 0.1, 최소 주제영역 지지도는 0.6, 최소 문서 지지도는 0.2, 특정 문서 지지도는 0.5라고 가정한다.

<표 1> 단어 "학교"에 대한 문서 지지도

영 역	S_1	S_2	**S_3**	S_4	S_5
문서지지도	0.2	0.2	**0.6**	0.3	0.3

<표 2> 단어 "학교"를 위한 불용어 해싱버킷 생성의 예

버 킷	B_0	B_1	**B_2**	**B_3**	B_4	B_5	**B_6**	B_7	B_8	B_9
주제영역 지지도	0	0	**0.4**	**0.4**	0	0	**0.2**	0	0	0
문서 지지도 평균	0.0	0.0	**0.2**	**0.3**	0.0	0.0	**0.6**	0.0	0.0	0.0
영 역			**S_1, S_2**	**S_4, S_5**			**S_3**			

주제영역 지지도가 높고 동시에 포함하고 있는 영역에서의 "학교" 단어의 문서 지지도의 평균이 가장 높은 버킷은 B_3이다. δ는 0.1이기 때문에 버킷 B_2와 B_3을 하나로 통합한다. 이때 통합된 버킷의 주제영역 지지도는 0.8, 문서지지도의 평균은 0.25가 된다. 이 버킷의 주제영역 지지도가 최소 주제영역 지지도 보다 크고, 문서 지지도가 최소 문서 지지도 보다 크므로 "학교"라는 단어는 불용어 후보 단어가 된다. 그러나, 영역 S_3에서 문서 지지도가 특정 문서 지지도 보다 크므로 영역 S_3에서는 특수한 용어로 사용되었다고 판단한다. 따라서, 영역 S_3을 제외한 나머지 영역 S_1, S_2, S_4, S_5에서 "학교"라는 단어를 제거한다.

3.2 단어가중치 산출 및 주제어 선정

불용어를 제거한 후 각 문서에 남아있는 단어들을 대상으로 단어가중치를 계산하여 해당문서를 대표하는 주제어들을 추출한다. 주제어를 추출하기 위한 단어가중치는 문서 내에서의 단어 빈도수(Term Frequency)로 계산되어진다[19]. 문서에서 추출된 단어의 중요도를 반영하기 위한 방법으로 TF×IDF(Term Frequency Inversed Document Frequency) 함수[4]가 가장 널리 사용되고 있다. TF×IDF 함수는 단어의 빈도수와 역 문서 빈도수를 곱하는 것으로 문서 d_i에서 단어 t_j의 가중치 $tfidf_{ij}$는 식 (4)와 같다.

$$tfidf_{ij} = tf_{ij} \times \ln \frac{N}{df_j} \tag{4}$$

이때 N은 전체 문서의 개수를 나타내고, tf_{ij}는 단어 빈도수로서 문서 d_i에서 단어 t_j가 나타난 횟수를 나타내며, df_j는 문서 빈도수로서 N개의 문서들 중에서 단어 t_j가 존재하는 문서수이다. 이 함수는 하나의 문서에 단어가 많이 포함되어 있다면 그 문서를 대표하는 단어로 사용될 가능성이 높지만 단어가 포함되어 있는 문서가 많을수록 문서를 특정 짓는 능력이 낮아진다는 것을 의미한다. 그러나, TF×IDF 함수는 처리하고자 하는 문서수가 많을수록, 즉 N값이 클수록 역문서 빈도수 값이 단어의 가중치를 결정하는데 많은 비중을 차지하는 단점을 가지고 있다. 일반적으로 한 문서 내에서 하나의 단어가 갖는 빈도수는 일정 범위를 유지하지만 역문서 빈도수 값은 $[0, \ln N]$의 범위를 가지게 되므로 N의 값이 클 경우 단어가중치는 N에 좌우되기 때문에 역문서 빈도수 값을 단어 빈도수 값의 범위와 유사하게 조절할 필요가 있다. 따라서, 본 논문에서는 역문서 빈도수 값의 최대값이 일정범위 μ이내가 되도록 정규화한 TF×NIDF(Term Frequency Normalized Inversed Document Frequency) 함수를 제안한다.

TF×NIDF 함수에서 단어 t_j의 역 문서 빈도수 idf_j를 $idf_j = \ln \frac{N}{df_j} = \ln N - \ln df_j$라고 했을 때, idf_j를 정규화한 $nidf_j$가 $[0, \mu]$가 되도록 조절한다. 이를위해 $nidf_j$는 idf_j와 마찬가지로 $nidf_j = \mu - \ln y$와 같은 형태로 나타내고, df_j가 $[1, N]$일 때, $\ln y$가 $[0, \mu]$가 되도록 y를 df_j에 관한 1차 함수로 나타낸다. 즉, df_j가 1일 때, y는 1이고, df_j가 N일 때, y는 e^{μ}이므로, $y = \frac{e^{\mu}-1}{N-1}(df_j - 1) + 1$이다. 따라서, 문서 d_i에서 단어 t_j의 가중치 $tfnidf_{ij}$는 식 (5)와 같다.

$$tfnidf_{ij} = tf_{ij} \times \left\{ \mu - \ln\left(\left(\frac{e^{\mu}-1}{N-1} \right)(df_j - 1) + 1 \right) \right\} \tag{5}$$

본 논문에서는 TF×NIDF 함수를 사용하여 문서에 나타나는 모든 단어의 가중치를 계산하고 각 단어의 중요도를 파악한다. 이들 단어들 중에서 가중치가 높은 주제어를 선별하여 문서 클러스터링을 수행할 때 주제어만을 대상으로 연산을 수행하면, 문서 클러스터링 수행 시간을 단축시킬 수 있으며, 문서 클러스터링 결과가 보다 더 확실해지는 이점이 있다. 앞 절에서 소개한 불용어 제거 방법이 문서 집합 전체에 대한 단어제거 방법이라고 보면, 문서에서 주제어를 선별하는 작업은 문서단위의 지역적인 방법으로 볼 수 있다. 문서의 주제어를 선별하기 위해 문서에 나타나는 모든 단어들의 가중치 평균값을 구하여 단어가중치가 평균값 이상인 단어들을 주제어로 선정한다.

또한 주제어를 선정한 후, 주제어들의 가중치에 대해 문서 길이 정규화 작업을 수행한다. 문서 길이 정규화를 수행하는 이유는 길이가 긴 문서의 경우 동일 단어가 반복해서 사용되는 경우가 많기 때문에 식 (5)의 단어가중치에서 단

어빈도수 값이 커지므로 문서 클러스터링에서 큰 비중을 차지하게 된다. 또한 길이가 긴 문서는 짧은 문서에 비해 더 많은 단어를 포함하고 있기 때문에 문서 클러스터링을 수행할 때 다른 문서들과 클러스터로 결합될 확률이 증가하게 된다. 한 문서 안에서 두 단어의 중요도는 단어들의 가중치를 절대적으로 비교하여 판별할 수 있지만 서로 다른 문서에서 추출된 단어의 중요도는 나타나는 문서의 길이에 따라 단어의 빈도수가 크게 차이날 수 있으므로 길이가 다른 문서에 대해 단어가중치를 절대적으로 비교하는 것은 의미가 없다. 따라서, 길이가 짧은 문서와 긴 문서간의 단어반복 빈도수의 격차를 해소하고, 문서 길이의 차이에 따른 가중치의 불균등을 해결하기 위해 각 문서의 길이에 따라 문서에 대한 단어가중치를 조정하는 문서 길이 정규화가 필요하다.

문서 길이 정규화를 위해 주로 사용되는 다양한 방법들은 다음과 같이 같다[13]. 첫째, 최대 빈도수 정규화(Maximum Frequency Normalization) 방법으로 문서에서 가장 많이 나타나는 단어의 빈도수로 각 단어의 빈도수를 나눠주는 방법이다. 둘째, 코사인 정규화(Cosine Normalization) 방법으로 여러 특성 정보로 구성된 벡터 공간 모델에서 가장 많이 사용되는 방법으로 벡터 W가 $W=\{w_1, w_2, \cdots, w_n\}$와 같을 때, 벡터의 각 원소를 코사인 정규화 원소인 $\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2}$로 나눠주는 것이다. 코사인 정규화는 높은 단어 빈도수에 대해서 정규화할 뿐만 아니라 단어 수가 많은 경우 코사인 정규화 원소가 증가하므로, 많은 단어에 대해서도 정규화가 가능하다. 따라서, 본 논문에서는 문서 길이 정규화를 위해 코사인 정규화 식을 사용하여 문서 d_i에서 주제어 k_j가 갖는 주제어 가중치를 식 (6)과 같이 정규화한다.

$$w(d_i, k_j) = \frac{tfnidf_{ij}}{\sqrt{\sum_{k=1}^{n} tfnidf_{ik}^2}} \quad (6)$$

3.3 초기 문서 클러스터링

초기 문서 클러스터링은 문서 d_i에서 주제어 k_j가 갖는 코사인 정규화된 가중치 $w(d_i, k_j)$가 주어졌을 때 유사 문서들의 집합인 문서 클러스터들을 찾는 방법이다.

[정의 3] 문서 유사도

문서에 포함되어 있는 주제어 및 주제어의 가중치를 기반으로 문서 d_i와 d_j간의 *유사도* $s(d_i, d_j)$는 식 (7)과 같이 정의된다.

$$s(d_i, d_j) = \frac{1}{2}\left(\frac{\sum_{k_l \in d_i \cap d_j} w(d_i, k_l)}{\sum_{k_l \in d_i} w(d_i, k_l)} + \frac{\sum_{k_l \in d_i \cap d_j} w(d_j, k_l)}{\sum_{k_l \in d_j} w(d_j, k_l)} \right) \quad (7)$$

계층적 집적 클러스터링[6]에서는 문서간의 유사도를 통해 클러스터간의 유사도를 단일 연결 방법, 완전 연결 방법 및 집단 평균 연결 방법으로 계산한다. 클러스터간의 유사도는 두 클러스터가 결합 가능한지를 판단할 때 적절한 척도로 사용될 수 있으나 유사도가 특정 클러스터에 있는 문서들의 유사정도를 나타내지 못한다. 따라서, 클러스터링 과정에서 두 클러스터의 결합 가능성 유무를 판단하고 동시에 클러스터에 있는 문서들의 유사정도를 나타낼 수 있는 척도로 본 논문에서는 클러스터의 *응집도*를 사용하며 다음과 같이 정의한다.

[정의 4] 클러스터 응집도

클러스터의 *응집도*는 클러스터에 소속된 문서들 간의 유사밀도를 나타내는 척도로써, 클러스터에 포함된 문서들이 서로 어느 정도 유사한지를 나타낸다. 따라서, 클러스터의 응집도는 클러스터에 포함된 문서들간의 유사도의 평균으로 표현하며, 클러스터에 포함된 문서들이 클러스터의 핵심 주제어들을 중심으로 얼마나 밀집되어 있는가를 나타낸다. 특정 클러스터 C_u의 *응집도* $c(C_u)$는 클러스터 C_u에 포함된 문서간의 유사도를 기반으로 하여 식 (8)과 같이 계산한다.

$$c(C_u) = \frac{\sum_{d_i \in C_u} \left(\sum_{d_j \in C_u - \{d_i\}} s(d_i, d_j) \right)}{|C_u| C_2} \quad (8)$$

이때 $|C_u|$는 클러스터 C_u에 포함되어 있는 전체 문서의 개수를 나타내며, 클러스터의 응집도는 임의의 서로 다른 문서 d_i, d_j의 문서 유사도 $s(d_i, d_j)$의 합을 전체 문서 중에서 임의의 문서 d_i, d_j를 선택하는 경우의 수로 나눈값으로 표현된다.

문서는 읽는 사람의 관점에 따라 문서의 주제가 조금씩 상이할 수 있으며, 다양한 주제를 포함할 수 있다. 따라서, 특정 클러스터가 일관된 주제에 해당하는 문서들로 구성되어 있는지에 대한 정확한 판단은 매우 어려운 문제이다. 따라서, 클러스터의 응집도만을 이용하여 문서 클러스터링을 수행할 경우 클러스터에 포함된 문서들이 어느 정도 밀접하게 연관되어 있는지를 확인할 수 있는 장점이 있으나 점진적 문서 클러스터링 과정에서는 전혀 상관이 없는 두 클러스터가 결합되어도 어느 정도의 응집도를 갖게 된다는 것이다. 즉, 크기가 크면서 응집도가 매우 높은 클러스터에 전혀 관련이 없는 작은 클러스터가 결합되었을 경우, 응집도가 높은 클러스터의 영향으로 이 두 클러스터를 결합한 클러스터의 응집도 역시 높게 된다. 이러한 문제점을 방지하기 위해 각 문서 기준으로 두 클러스터의 유사도를 나타내는 척도로 두 클러스터를 통합할 때 두 클러스터간의 참여도를 다음과 같이 정의한다.

[정의 5] 클러스터 참여도

두 개의 임의의 클러스터 C_m과 C_n이 있을 때, 클러스터 C_n에 대한 클러스터 C_m의 참여도 $p(C_m|C_n)$는 식 (9)과 같다.

$$p(C_m|C_n)=\frac{\sum_{d_i\in C_m}\left(\sum_{k_l\in CK_m\cap CK_n} w(d_i,k_l)\right)}{\sum_{d_i\in C_n}\left(\sum_{k_l\in CK_n} w(d_i,k_l)\right)} \quad (9)$$

이때 CK_n은 클러스터 C_n에 속하는 주제어의 집합을 의미하며 클러스터의 참여도는 두 개의 클러스터의 주제어 집합에 공통으로 포함되어 있는 주제어 가중치의 합을 클러스터 C_n의 주제어 가중치의 합으로 나눈 값으로 표현한다. 따라서, 두 개의 별도 척도인 참여도와 응집도로 두 개의 유사 클러스터 결합 가능성을 다음과 같이 정의한다.

[정의 6] 클러스터 결합 가능성

두 개의 클러스터 C_m과 C_n이 다음에 주어진 두 가지의 *클러스터 결합 조건*을 만족할 때, 두 클러스터는 결합이 가능하고 두 클러스터를 결합한 새로운 클러스터 $C_m \vee C_n$은 다음을 만족한다.

*MinPart*를 최소 클러스터 참여도라고 하고, *MinCoh*를 최소 클러스터 응집도라고 할 때

1) $p(C_m|C_n) \geq MinPart$ *and* $p(C_n|C_m) \geq MinPart$
2) $c(C_m \vee C_n) \geq MinCoh$ ■

계층적 집적 클러스터링 방법은 문서간의 유사도 및 클러스터간의 유사도를 이용하여 클러스터링을 수행하는 방법이다. 클러스터간의 유사도는 문서간의 유사도를 기반으로 계산하며, 클러스터링 병합 방법에는 단일 연결 방법(Single Link Method), 완전 연결 방법(Complete Link Method), 그리고 집단 평균 연결 방법(Group Average Link Method)이 보편적으로 사용된다[6]. 단일 연결 방법은 두 클러스터가 있을 때, 각 클러스터에 포함되어 있는 문서들간의 유사도 중에서 최대 유사도를 클러스터의 유사도로 정한다. 따라서, 두 클러스터가 유사한 문서를 포함하고 있다면, 유사한 클러스터로 간주한다. 완전 연결 방법은 단일 연결 방법과는 반대로 각 클러스터에 포함되어 있는 문서들간의 유사도 중에서 최소 유사도를 클러스터의 유사도로 정하는 방법이다. 즉, 클러스터에 포함되는 문서들이 모두 유사할 때 클러스터도 유사한 것으로 본다. 집단 평균 연결 방법은 클러스터의 유사성을 결정하기 위해 각 클러스터에 포함된 문서간의 유사도의 평균값을 클러스터의 유사도로 이용한다. 그러므로, 이 방법은 단일 연결 방법과 완전 연결 방법의 중간적인 구조를 나타낸다. 이 중에서 완전 연결 방법이 클러스터에 포함된 모든 문서들이 유사하기 때문에, 다른 방법에 비해 크기가 작고, 단단하게 결합된 클러스터들을 생성한다[16].

문서집합 D에 대한 계층적 문서 클러스터링은 클러스터의 최소 응집도와 최소 참여도에 기반하여 다음의 과정에 따라 직접 클러스터링을 수행한다.

단계 1 : 주어진 문서 집합 D에서 하나씩의 문서를 갖는 독립된 클러스터를 생성한다.

단계 2 : 클러스터들 중에서 클러스터 결합 조건을 만족하면서 결합했을 때 응집도가 가장 높은 두 클러스터를 찾아서 결합한다.

단계 3 : 더 이상 최소 응집도와 최소 참여도를 만족하지 않을 때까지 단계 2의 과정을 계속 반복한다.

4. 점진적 문서 클러스터링

계층적 집적 클러스터링 방법은 이전의 클러스터링 정보를 활용하지 못하고, 클러스터링을 수행한 후에 새로운 문서들이 추가될 경우 처음부터 다시 모든 문서에 대해 클러스터링을 수행해야 하는 단점이 있다. 이러한 방법은 클러스터링을 수행하지 않아도 되는 문서들을 다시 클러스터링하므로 시간을 낭비하고, 불필요한 클러스터링 작업을 반복적으로 수행할 수 있다. 따라서, 본 장에서는 3장의 초기문서 클러스터링 결과를 기반으로 새로 추가된 문서들에 대해서 점진적인 방법으로 문서 클러스터링을 수행할 수 있는 방법을 제시한다. 점진적 문서 클러스터링 방법은 초기 문서 클러스터링 방법을 이용하여 생성된 클러스터들을 기반으로 새로 추가된 문서에 대해 적합한 클러스터를 찾아 문서를 할당하거나 새로운 클러스터를 점진적으로 생성하는 방법이다. 새로운 문서에 대해 전체 클러스터에서 적합한 클러스터를 찾는 것은 클러스터의 수가 많을수록 많은 탐색시간이 필요하다. 따라서, 본 논문에서는 현재의 클러스터들에 대한 카테고리 트리를 생성하여 클러스터들을 구조화하여 문서가 추가될 때 문서에 적합한 클러스터를 효과적으로 검색할 수 있도록 한다.

4.1 카테고리 트리 생성

카테고리 트리는 현재 클러스터링된 문서의 주제어들로 구성된 트리로 특정 주제 카테고리는 자신의 카테고리의 주제어 집합과 하위카테고리들의 집합 및 카테고리에 소속되는 클러스터들의 집합으로 구성된다. 카테고리 트리는 카테고리를 노드로 하여 클러스터들의 계층적 구조를 나타내는 일종의 트리 구조이다. 카테고리 트리를 생성하는 방법의 핵심은 리프노드인 각 클러스터들의 중심 주제어들을 독립된 문서로 간주하여 문서집합을 만든 후 3.2절에서 기술된 방법을 적용하여 문서 집합의 주제어를 상위 카테고리로 생성한다. 이러한 과정을 반복적으로 적용하여 상향식으로 카테고리 트리를 생성하게 되며, 카테고리 트리를 생성하는 구체적인 단계는 다음과 같다.

단계 1 : 각각 독립된 클러스터들을 모두 리프노드로 설정한다.

단계 2 : 리프노드의 클러스터들을 문서로 간주하여 주제어들을 선택한다. 즉, 클러스터의 주제어들을 문서의 주제어와 같게 생각한다.

단계 3 : 코사인 정규화 공식을 사용하여 정규화한다.

단계 4 : 3.3절에서 기술된 방법으로 클러스터링을 과정을 수행한다.

단계 5 : 문서 집합의 주제어들을 상위 카테고리로 생성한다.

단계 6 : 더 이상 클러스터링 과정이 이루어지지 않을 때까지 단계 4와 단계 5의 과정을 반복한다.

단계 7 : 클러스터링 과정이 종료된 후에 현재 만들어진 카테고리들을 임의의 루트 카테고리를 생성하여 연결한다. 이때 클러스터링 과정이 이루어지지 않은 클러스터들은 리프노드로 남는다.

단계 8 : 전체 문서의 주제어 가중치를 더하여 루트 카테고리의 주제어 가중치를 설정한다.

이때 카테고리의 주제어 가중치는 클러스터에 포함되어 있는 문서들의 주제어 가중치로 표현된다.

4.2 문서의 삽입 및 삭제

카테고리에 새로운 문서를 삽입하기 위해서는 새로운 문서가 카테고리에 어느 정도 참여할 수 있는지를 파악해야 한다. 이를 카테고리 참여도라고 하며, 카테고리 참여도는 주어진 카테고리에 새로운 문서의 삽입여부를 결정하는 척도로 사용되어지며 다음과 같이 정의한다.

[정의 7] 카테고리 참여도

문서 d와 카테고리 T가 있을 때, 카테고리 T에 대한 문서 d의 참여도 $p(d|T)$는 식 (10)과 같다.

$$p(d|T)=\frac{\sum_{k_i\in(TK\cap d)} w(k_i)}{\sum_{k_i\in TK} w(k_i)} \tag{10}$$

이때 k_i는 카테고리 T의 주제어를 의미하며, $w(k_i)$는 주제어 k_i의 가중치를 나타낸다. 또한 TK는 카테고리 T의 주제어 집합을 나타낸다.

새로운 문서 d를 카테고리 트리에 삽입할 때, 문서 d가 삽입될 클러스터를 카테고리 트리에서 먼저 탐색한다. 그 후, 새로운 문서 d의 주제어를 3.2절에서 기술한 방법으로 추출한 후 카테고리 트리의 각 노드들의 주제어들과 비교하여 카테고리 참여도가 주어진 최소 참여도보다 큰 카테고리를 선택한다. 카테고리 트리의 루트 카테고리에 대한 문서 d의 참여도를 계산하여 카테고리 최소 참여도 보다 작으면 문서 d의 클러스터 대상에서 루트 카테고리를 제외한다. 그러나, 루트 카테고리에 대한 문서 d의 카테고리 최소 참여도가 카테고리 최소 참여도 보다 크다면 새로운 문서가 기존의 클러스터에 흡수되거나, 흡수된 문서에 의해 클러스터가 합병되는 경우는 클러스터 결합 조건과 문서의 카테고리에 대한 참여도를 이용하여 문서가 삽입될 클러스터를 탐색하여 클러스터 결합 조건을 만족하는 클러스터들 중에서 가장 응집도가 높은 클러스터에 문서 d를 추가한다. 이와 같이 적합한 클러스터를 탐색하여 문서를 추가한 후에, 문서가 추가된 클러스터와 같은 카테고리에 포함된 클러스터들을 다시 클러스터링한다. 왜냐하면 동일 카테고리의 서브클러스터들은 그만큼 유사한 주제어들을 갖고 있으므로, 다른 카테고리에 포함된 클러스터들보다 병합될 수 있는 가능성이 크기 때문이다.

이때 어떤 문서 클러스터에도 삽입되지 않은 문서는 노이즈 문서가 된다. 카테고리에 문서를 삽입할 때 새로 추가된 문서와 노이즈 문서들에 대해서 비교를 하지 않기 때문에 문서가 계속적으로 추가될 때마다 노이즈도 증가하게 된다. 따라서, 노이즈 문서들에 대해서 문서 클러스터링을 수행하여 새로운 클러스터를 생성하여야하나 문서가 추가될 때마다 노이즈 문서들을 클러스터링하는 것은 매우 비효율적일 뿐만 아니라, 새로운 클러스터가 생성될 확률이 낮기 때문에 노이즈 문서가 일정량에 도달하였을 때, 클러스터링을 수행하는 것이 효율적이다. 노이즈 문서들을 클러스터링한 후에는 앞서 소개한 카테고리 트리 생성 방법을 통해 카테고리 트리를 생성한다.

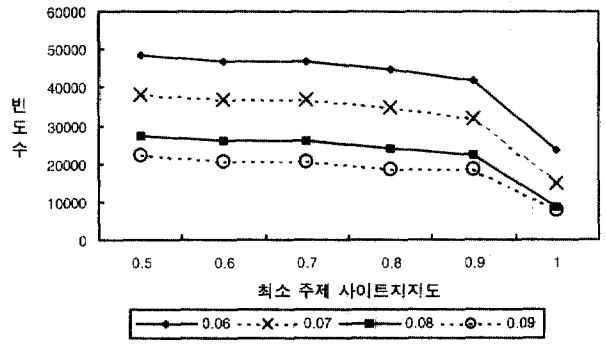
기존의 클러스터에 포함되어 있거나, 혹은 노이즈인 문서는 문서가 제거되어도 카테고리 트리에는 아무런 영향을 미치지 않으나, 클러스터에 포함된 문서가 삭제되었을 경우 문서가 삭제된 이후에도 클러스터가 최소 응집도를 만족시키면 클러스터를 분해하지 않는다. 이와 반대로 클러스터에 포함된 문서가 제거된 후 클러스터의 응집도가 최소 응집도보다 작으면 클러스터를 분할한다. 이때 문서가 삭제된 이후 클러스터에 남아있는 문서들을 다시 클러스터링하여 클러스터들을 생성하고 카테고리의 하위 클러스터에 추가한다. 그러나, 클러스터가 생성되지 않은 경우 남아있는 문서들을 모두 노이즈로 처리한다.

5. 실험 및 결과 분석

본 장에서는 본 논문에서 제안하고 있는 불용어 제거 알고리즘 및 계층적 집적 클러스터링 알고리즘과, 점진적 문서 클러스터링 알고리즘으로 수행한 실험에 관하여 기술한다. 실험에 사용된 데이터는 야후! 코리아 뉴스[20]에서 제공하고 있는 신문기사 중에서 경제, IT, 정치, 사회 등 10개 영역(사이트)의 기사를 웹 로봇 에이전트를 사용하여 추출하였다. 추출된 영역별 평균 문서수는 1026개이고, 총 819835개의 명사를 포함한다. 추출된 기사에서 형태소 분석기인 HAM[3]을 이용하여 112860개의 명사를 추출하였다.

가장 먼저 추출된 명사를 대상으로 불용어 제거 알고리즘을 이용하여, 불용어 제거 실험을 수행하였다. 이 실험에 사용된 대부분의 단어들의 각 영역별 문서 지지도를 조사하였을 때, 대부분 단어의 문서 지지도가 0.3 이하를 기록하여, 특수 용어 판별을 위한 특정 문서 지지도를 0.3, 최소 문서지지도를 0.1 이하로 설정하였다. (그림 1)은 최소 주제 영역 지지도와 최소 문서 지지도에 따라서 제거된 불용어

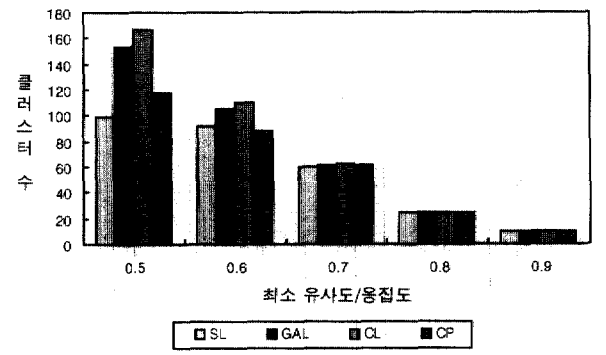
들의 빈도수를 나타낸 것이다. x축은 최소 주제영역 지지도이며, 최소 문서 지지도가 0.06, 0.07, 0.08, 0.09 일 때 불용어로 판별된 단어들의 빈도수를 나타낸다.



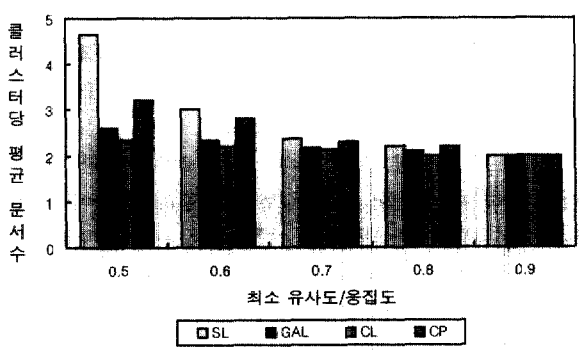
(그림 1) 불용어 빈도수

(그림 1)에서 알 수 있듯이 동일한 최소 문서 지지도를 가질 때 최소 주제영역 지지도가 높을수록 불용어들의 비율이 낮아지며, 동일한 최소 주제영역 지지도를 가질 때는 최소 문서 지지도가 높을수록 불용어들의 비율이 낮아지는 것을 확인할 수 있다.

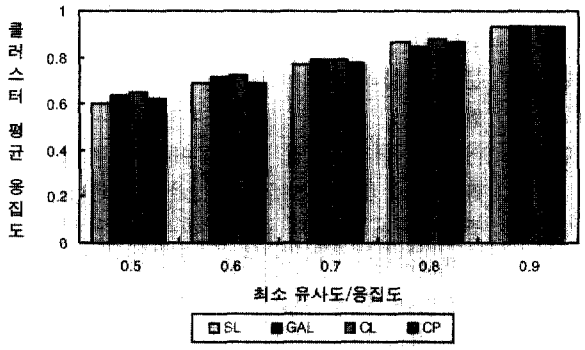
(그림 2)는 계층적 집적 클러스터링 방법에서 유사도 측정을 위한 세 가지 방법-단일 연결 방법(SL), 집단 평균 연결 방법(GAL), 완전 연결 방법(CL)-과 본 논문에서 사용하고 있는 응집도와 참여도를 이용한 방법(CP)을 비교하는 실험을 수행하여 각 클러스터의 비교 결과를 나타낸다.



(a) 클러스터 수



(b) 클러스터당 평균 문서수



(c) 클러스터 평균 응집도

(그림 2) 계층적 문서 클러스터링 비교

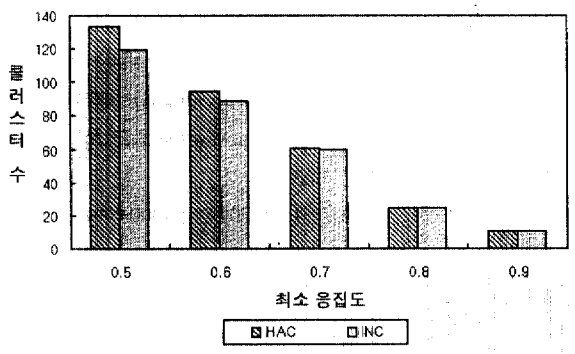
각 방법에 의해 클러스터링을 수행한 후, 최소 유사도와 최소 응집도에 대한 클러스터의 수와 클러스터당 평균 문서수, 클러스터의 평균 응집도를 비교하였다. 클러스터들의 평균 응집도는 SL, GAL, CL, CP가 비슷한 정도를 나타냈으며, SL 방법이 클러스터의 수는 적고, 대신 클러스터당 평균 문서수가 높게 나타난 것에 비하여, GAL, CL 방법은 클러스터의 수가 많고, 클러스터당 평균 문서수가 낮게 나타났다. 그러나 응집도와 참여도를 사용한 클러스터링 방법은 응집도는 비슷하면서 적정 크기의 클러스터를 적정 개수만큼 생성하는 것으로 판단할 수 있다.

다음은 계층적 집적 클러스터링 방법(HAC)에 의해 생성된 클러스터와 점진적 문서 클러스터링 방법(INC)에 의해 생성된 클러스터의 특성에 대한 비교실험을 나타낸다. 이 실험에서는 HAC과 INC에 의해 생성되는 클러스터들의 개수와 클러스터당 평균 문서수, 클러스터들의 평균 응집도를 비교하고, INC에 의해 생성되는 클러스터들의 HAC에 의해 생성된 클러스터에 대한 오차를 계산한다. 그리고, 문서수를 증가시키면서 HAC와 INC에 의해 클러스터링이 수행되는 시간을 비교한다. HAC에 의해 생성된 클러스터들의 집합을 $HC=\{C_1, C_2, \cdots, C_n\}$이라 하고, INC에 의해 생성된 클러스터들의 집합을 $IC=\{C_1', C_2', \cdots, C_m'\}$라고 할 때, IC의 오차는 HC와 IC의 유사도를 통해 계산한다. IC에 포함된 클러스터 C_i'의 HC에 대한 유사도는 HC의 클러스터들 중에서 C_i'에 포함된 문서를 가장 많이 포함하고 있는 클러스터 C_j를 찾아 공통된 문서의 비율을 $sim(C_i', HC)$라고 한다. 따라서, $sim(C_i', HC)=\max\left(\left\{\frac{|C_j \cap C_i'|}{|C_i'|} \mid C_j \in HC\right\}\right)$이다. 또한 HC와 IC의 유사도 $sim(IC, HC)=\frac{\sum_{C_i' \in IC} sim(C_i', HC)}{|IC|}$와 같이 계산한다. 따라서, HC와 IC의 오차 ε는 식 (11)과 같다.

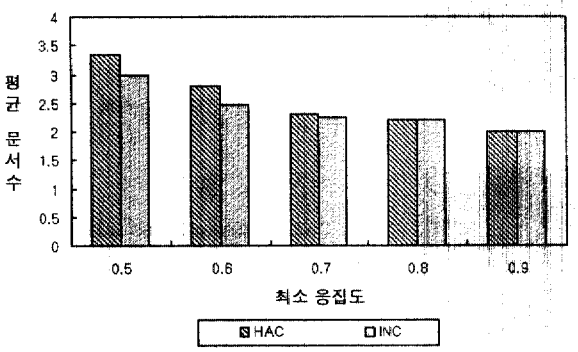
$$\varepsilon = 1 - sim(IC, HC) \tag{11}$$

이 실험에서 HAC는 중복을 허용하지 않기 때문에, INC

에서도 중복을 허용하지 않는 방법을 사용하여 경제 분야의 문서 1000개에 대해서 *MinPart*(최소 참여도)가 0.2일 때, 동일한 *MinCoh*(최소 응집도)에 대해 실험을 수행하였다. (그림 3)은 HAC와 INC에 의해 최종적으로 생성된 클러스터 수와 클러스터들의 평균 문서수를 비교한 것이다. 실험 결과, 최소 응집도가 낮을 때에는 INC에 의해 클러스터링을 수행한 경우, 클러스터의 수가 적고, 생성된 클러스터의 크기가 작지만, 응집도가 높을수록 결과가 거의 같아지는 것을 알 수 있다.



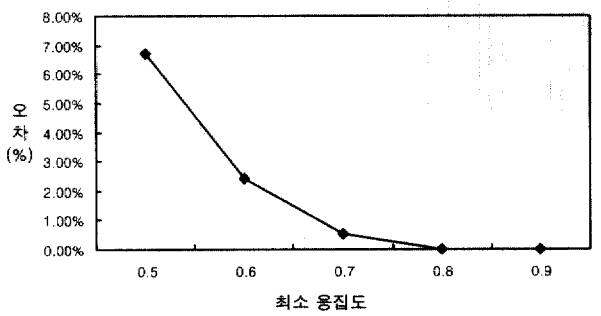
(a) 클러스터 수



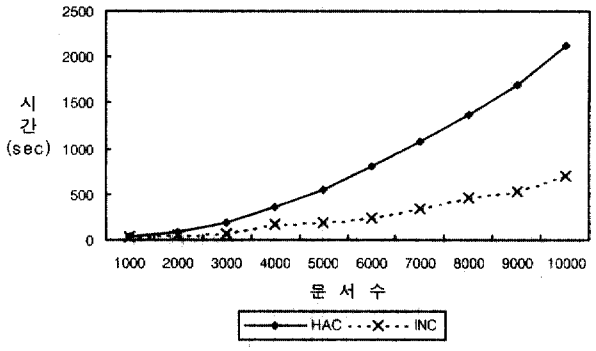
(b) 평균 문서수

(그림 3) HAC와 INC의 클러스터 수 및 평균 문서수

(그림 4)는 동일한 최소 응집도에서 HAC와 INC로 클러스터링을 수행하였을 때, 생성된 클러스터들의 오차를 계산한 것이다. 최소 응집도가 높을수록 오차가 적게 나타나는 것을 확인할 수 있다.



(그림 4) HAC와 INC의 오차



(그림 5) HAC와 INC의 수행 시간

마지막으로 문서수의 증가에 따른 HAC와 INC의 수행시간을 비교하였다. n 개의 초기 문서에 대해서 m 개의 문서가 추가된다고 할 때, HAC에 의한 수행 시간은 $(n+m)$개의 문서를 클러스터링하는 시간이 되며, INC에 의한 수행 시간은 m 개의 문서를 카테고리 트리에 추가하는데 걸리는 시간이 된다. (그림 5)는 *MinPart*와 *MinCoh*를 각각 0.2와 0.5로 설정하였고, n 과 m 은 모두 1000이라고 가정하였을 때, HAC와 INC에 의한 수행시간을 나타낸 것이다. (그림 5)에서 보듯이 문서수가 증가할수록 HAC와 INC의 수행 속도가 급격하게 차이가 나는 것을 확인할 수 있다.

6. 결론 및 향후 연구

본 논문에서는 불용어 제거 알고리즘을 통해 문서 클러스터링에서 불필요한 단어를 찾아내어 제거함으로써 문서 클러스터링의 정확도를 높인다. 따라서, 주제어의 단어가중치 산출을 위해서 TF×IDF 공식이 갖는 약점을 극복한 TF×NIDF 공식을 제안하여 계층적 문서 클러스터링에 사용하여 클러스터들의 카테고리 트리를 작성함으로써, 사용자가 검색하고자 하는 문서를 주제영역별로 다양한 접근 경로를 통해 탐색하는 것이 가능할 뿐만 아니라, 새로 문서가 추가될 때 다시 문서 클러스터링을 수행하지 않고, 추가된 문서만 카테고리 트리에 삽입하는 점진적 문서 클러스터링 방법을 제안하였다. 본 논문에서 제안하는 불용어 제거 알고리즘은 클러스터링을 먼저 수행하고, 클러스터링된 문서들을 토대로 불용어 제거 알고리즘을 수행하여 불용어를 판별함으로써 불용어 제거 알고리즘의 유용성을 증명하였으며, 문서 클러스터링 실험에서는 기존의 HAC 알고리즘과 본 논문에서 제안하는 HODC 알고리즘을 INC 알고리즘과 비교하여, 문서가 증가할 때 INC 알고리즘에 의해 생성된 클러스터가 HAC 알고리즘에 의해 생성되는 클러스터와 유사하면서, INC 알고리즘이 HAC 알고리즘보다 빠른 수행 속도를 보이는 것을 확인하였다. 점진적 문서 클러스터링 방법이 주기적으로 노이즈 문서들을 별도로 재클러스터링을 수행해야 하기 때문에 최소 응집도를 높게 설정한 경우 노이즈 문서들의 재클러스터링에 많은 시간이 걸릴 수 있

으므로 향후 새로운 문서가 추가될 때, 노이즈 문서들을 빠른 시간에 재클러스터링할 수 있는 기법에 대한 연구가 수행되어야 한다.

참 고 문 헌

[1] Douglass, R. Cutting, David, R. Karger, Jao, O. Pedersen, and John, W. Tukey, "Scatter/Gather : A Cluster-based Approach to Browsing Large Document Collections," *15th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.318-329, 1992.

[2] B. W. Frakes and R. Baeza-Yates, "Information Retrieval : Data Structures & Algorithms," Prentice Hall, 1992.

[3] 강승식, "HAM : 한국어 분석 모듈", http://nlp.kookmin.ac.kr.

[4] G. Salton, C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, Vol.24, No.5, pp.513-523, 1988.

[5] "야후!", http://www.yahoo.com/.

[6] Jain, A. K. and Dubes, R. C., "Algorithms for Clustering Data," Prentice Hall, 1988.

[7] J. J. Rocchio, "Document Retrieval Systems - Optimization and Evaluation," Ph. D. Thesis, Havard University, 1966.

[8] C. J. Van Rijsvergen, "Information Retrieval," Butterworth, London, 2nd edition, 1979.

[9] David D. Lewis, Robert E. Schapire, James P.Callan, Ron Papka, "Training Algorithms for Linear Text Classifiers," *Proceedings of 19th ACM International Conference on Research and Development in Information Retrieval*, 1996.

[10] Eui-Hong (Sam) Han, George Karypis, and Vipin Kumar, "Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification," *5th Pacific Asia Conference on Knowledge Discovery And Data Mining*, 2001.

[11] Yiming Yang, "Expert Network : Effective and efficient learning from human decisions in text categorization and retrieval," *17th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.13-22, 1994.

[12] Ron Fagin, Yoelle Maarek, Israel Ben-Shaul, and Dan Pelleg, "Ephemeral document clustering for web applications," IBM Research Report RJ 10186, April, 2000.

[13] Amit Singhal, Chris Buckley, and Mandar Mitra, "Pivoted Document Length Normalization," *Proceedings of 19th ACM International Conference on Research and Development in Information Retrieval*, 1996.

[14] M. Ester, H. Kriegel, J. Sander, M. Wimmer, and X. Xu, "Incremental Clustering for Mining in a Data Warehousing Environment," *Proceedings of the 24th VLDB Conference*, New York, USA, 1998.

[15] Futamura Shoukchi and Matsuo Fumihiro, "Automatic Indexing by Stop Word Removal on Scientific and Technical Documents Written in English," *Information Processing Society of Japan*, Vol.28 No.07, 1987.

[16] G. Salton, "Automatic Text Processing," Addison-Welsley Publishing Company, 1989.

[17] Weifeng Li, Baowen Xu, Cheng-Cheng Chu, Chih-Wei Lu, "Application of Genetic Algorithm in Search Engine," *Proceedings of International Symposium on Multimedia Software Engineering*, pp.366-371, 2000.

[18] W. E. L. Grimson and D. P. Huttenlocher, "On the sensitivity of geometric hashing," *3rd International Conference on Computer Vision*, pp.334-338, 1990.

[19] I. Aalbersberg, "A Document Retrieval Model Based on Term Frequency Ranks," *17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.163-172, 1994.

[20] 야후!코리아 뉴스, http://kr.dailynews.yahoo.com/.

강 동 혁

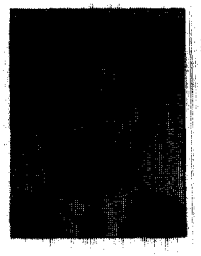
e-mail : wolfkang@netville.co.kr

2000년 연세대학교 기계전자공학부(공학사)

2002년 연세대학교 컴퓨터과학산업시스템 공학과 (공학석사)

2002년~현재 (주)네트빌 부설연구소 연구원

관심분야 : CASE, CBD, 텍스트 마이닝



주 길 홍

e-mail : faholo@amadeus.yonsei.ac.kr

1998년 인천대학교 전자계산학과(공학사)

2000년 연세대학교 컴퓨터과학과(공학석사)

2000년~현재 연세대학교 컴퓨터과학과 (박사과정)

관심분야 : 분산데이터베이스, 미디에이터 시스템, 분산질의처리, 질의최적화, 웹 데이타베이스마이닝

이 원 석

e-mail : leewo@amadeus.yonsei.ac.kr

1985년 미국 보스턴대학교 컴퓨터공학과 (공학사)

1987년 미국 퍼듀대학교 컴퓨터공학과 (공학석사)

1990년 미국 퍼듀대학교 컴퓨터공학과 (공학박사)

1990년~1992년 삼성전자 선임연구원

1993년~1999년 연세대학교 컴퓨터과학과 조교수

1999년~현재 연세대학교 컴퓨터과학과 부교수

관심분야 : 분산데이터베이스, 미디에이터시스템, 데이터마이닝, 침입탐지, 멀티미디어데이터베이스