웹 에이전트를 위한 통합방식 문서 클러스터링

(A Hybrid Document Clustering for a Web Agent)

양 찬 범 † 이 성 열 † 박 영 택 ††

(Chan-Bum Yang) (Sung-Yull Lee) (Young-Tack Park)

요 약 웹 에이전트는 사용자가 웹을 브라우징하는 행위를 모니터하여 사용자의 관심 정보를 학습하고 사용자가 필요로 하는 웹 상의 정보를 자동 제공하는 지능형 시스템이다. 웹 에이전트가 사용자의 선호도를 학습하기 위해서는 귀납적 기계학습을 수행하는데, 이때 학습의 효율을 높이기 위해서는 사용자가 관심있어하는 문서들을 유사한 문서들로 클러스터링하여 학습 시스템에 제공하여야 한다. 본 논문에서는 웹 에이전트의 학습 시스템에 입력되는 학습대상 문서들을 보다 정확하고 효율적으로 클러스터링하여 제공하기 위해서 Top-down 방식과 Bottom-up 방식을 통합 적용한 통합방식 문서 클러스터링과 초기 클러스터 생성을 위한 평가함수를 제시한다. Top-down 방식으로는 개념적 클러스터링 알고리즘인 COBWEB을 적용하고, Bottom-up 방식으로는 교차기반(Intersection-based) 클러스터링 방식인 Etzioni의 클러스터링 알고리즘을 적용하였다.

Abstract A web agent is an intelligent system which learns preferences of users by monitoring the behavior of users and provides the personalized search service to users. Because a web agent performs inductive learning to extract preference information of users, input documents submitted to inductive learning system need to be categorized. In this paper, we propose a hybrid document clustering method that integrates a top-down and a bottom-up clustering methods for efficient clustering of input documents. We apply COBWEB to top-down clustering, and apply intersection-based clustering of Etzioni's bottom-up method. COBWEB produce a document classification tree from input documents. We propose the evaluation function for creating the efficient initial cluster from the classification tree. We can reduce the count of inter-cluster merge operation by using the evaluation function. Ultimately, we will show that the proposed clustering method outperforms the pervious clustering method in the aspects of precision and execution speed.

1. 서 론

현재 인터넷상의 웹 문서의 수는 급격하게 증가하고 있다. 하루에도 수천의 홈페이지가 새로이 만들어지고 다량의 웹 문서들이 인터넷을 통하여 사용자에게 제공된다. 이러한 정보의 홍수 속에서 사용자는 자신의 요구에 적합한 정보를 찾기 위하여 많은 시간을 투자하여 웹을 검색하고 정보를 정리하는 일에 매달려야 한다. 웹 에이전트는 이러한 문제를 해결하기 위한 방법으로 사용자의 웹 브라우징 행위를 모니터링하여 사용자의 관심정보를 학습하고 각 사용자에게 적합한 정보를 제공하는 역할을 수행한다. 웹 에이전트는 사용자의 관심정보를 추출하기 위해서 사용자가 관심 있어 하는 웹 문서를 추출하여 학습시스템으로 제공하고 사용자 프로파일을 생성하게 된다. 이때 학습 시스템에 제공되는 사용자 관심문서는 상호 관련이 있는 클러스터링된 형태로 제공되어야 한다. 사용자가 관심을 보인 많은 문서들 중에서 사용자의 관심 정보를 추출하려면 먼저, 비슷한 내용들을 가지는 문서들을 서로 클러스터링 시킨 후 여기에서 사용자 관심정보를 추출하는 것이 효율적이기 때

* 본 연구는 과학재단 핵심전문연구과제(971-0901-009-2)의 지원을 받았습니다.

† 비 회 원 : 숭실대학교 컴퓨터학부
cbyang@muiti.soongsil.ac.kr
trident@agentxpert.com
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
park@computing.soongsil.ac.kr

논문접수 : 2000년 3월 24일
심사완료 : 2001년 3월 28일

문이다. 이러한 이유로 학습 시스템에 제공되는 사용자 관심문서는 상호 관련이 있는 클러스터링된 형태로 제공되어야만 사용자 관심정보 추출에 대한 학습 효율을 높일 수 있다[1][2]. 따라서 본 논문에서는 사용자 관심 문서 클러스터링시 문서 분류 트리에서 충분한 크기와 응집도를 가지는 초기 클러스터 생성을 위한 평가함수를 제안하여 기존의 클러스터링 방식보다 정확하고 효율적인 클러스터링 방식을 구현하는 것을 목적으로 한다.

본 논문에서는 이러한 문서 분류와 클러스터링 작업을 수행하기 위해서 Top-down 클러스터링 방식과 Bottom-up 클러스터링 방식을 통합한 통합방식 문서 클러스터링 방법을 제시한다. Top-down 클러스터링 방식은 개념적 클러스터링 알고리즘인 COBWEB을 적용하였다. COBWEB은 미 분류 문서집합을 입력집합으로 하여 문서들간의 유사도를 표현하는 문서 분류트리를 생성한다. 생성된 문서 분류트리는 본 논문에서 제안하는 평가함수에 의하여 초기 클러스터를 생성하고, 이는 다음 단계의 입력집합으로 사용된다. Bottom-up 클러스터링 방식은 Etzioni의 교차기반 클러스터링 알고리즘을 이용한다. 이는 초기 클러스터링된 문서들을 입력집합으로 하며 평가함수를 이용하여 클러스터간 유사도를 계산한 후에 보다 동일한 개념의 문서들을 포함하는 클러스터 병합 과정을 거쳐 최종 클러스터를 생성하게 된다. 본 논문의 구성은 다음과 같다.

2장에서는 웹 문서 클러스터링에 대한 관련연구를 살펴보고 3장에서는 COBWEB과 Etzioni를 이용한 Top-down, Bottom-up 방식의 통합 클러스터링 방식과 초기 클러스터 생성을 위한 평가함수에 대하여 자세히 설명한다. 4장에서는 초기 클러스터 생성을 위해 제시한 평가함수의 유용성과 실제 웹 문서들을 대상으로 한 통합방식 문서 클러스터링의 성능을 비교 설명하고 5장에서는 결론을 맺는다.

2. 관련 연구

근래에 들어 인터넷 사용자의 웹 검색을 도와주기 위한 여러 연구들이 활발히 진행되고 있다. 이 가운데 한 방법은 사용자가 관심 있어 하는 문서들을 비슷한 문서들로 분류하여 웹 검색 결과를 브라우징 하기 쉽도록 클러스터링 하여주는 방법이 있다. 다음은 웹 에이전트 분야에서 문서 클러스터링 방식을 사용한 관련연구를 살펴보겠다.

2.1 Grouper [Washington Univ.]

Washington 대학에서 만든 Grouper는 메타 검색 엔진의 인터페이스로서 검색 결과로 제공되는 짧은 단문(Snippet)을 입력으로 클러스터링을 수행하여 이 결과를 사용자에게 제공한다[3]. Grouper를 개발한 Etzioni는 메타 검색 엔진으로 유명한 Metacrawler를 개발했으며, Grouper는 Metacrawler에 기반한 메타 검색 엔진의 인터페이스 기능을 한다. 기존의 정보 검색 분야에서의 클러스터링과 비교해서 웹 검색 결과를 클러스터링하기 위해서는 다음과 같은 요구사항을 만족해야 한다.

- 클러스터링 결과를 브라우징하기 쉬워야 한다.
- 클러스터링 수행 속도가 빨라야 한다.
- 문서 수 증가에 따른 속도 저하가 없어야 한다.
- 문서 전처리 과정이 없어야 한다.
- 단문(Snippet)으로 클러스터링 가능해야 한다.

Etzioni팀은 이와 같은 요구사항을 만족하는 새로운 클러스터링 방식인 교차기반 클러스터링 방식을 소개했다. 교차기반 클러스터링방식은 GQF(Global Quality Function) 평가함수를 이용하는 단어-교차(word-intersection) 클러스터링 방식과 Suffix 트리를 이용하는 절-교차(phrase-inte- rsection) 방식으로 분류된다. 단어-교차 클러스터링 방식은 절-교차 클러스터링 방식보다 클러스터링 정확도 면에서는 우수하지만 속도 면에서는 더 많은 클러스터링 수행 시간을 필요로 한다.

2.2 WebACE[Minnesota Univ.]

Minnesota 대학에서 개발한 WebACE는 지능형 웹 검색 에이전트로서 클라이언트 쪽에 프락시 서버를 두어서 사용자의 관심 문서를 파악하고, 관심 문서들을 클러스터링 함으로써 사용자 프로파일을 구축한다[4]. 사용자 프로파일은 키워드와 값의 벡터 리스트 형태로 사용자 관심 문서를 클러스터링한 결과로부터 검색 질의어를 생성하여 웹 상에서 사용자가 관심 있어 할 만한 문서들을 검색하여 제공한다.

WebACE는 문서 클러스터링을 위해서 2가지 새로운 방식을 소개하고 있다. 하나는 ARHP(Association Rule Hypergraph Partitioning Algorithm)으로서 연관 규칙(Association rules)과 하이퍼그래프 분할(Hypergraph Partitioning)을 이용하여 슈퍼마켓의 바코드 데이타 등과 같은 트랜잭션 기반 데이타베이스 항목을 위한 클러스터링 방식이다.

문서 검색 도메인의 관점에서 보면, 각 문서는 키워드와 값의 벡터 리스트로 표현되므로 ARHP 방식은 문서 클러스터링에 적용될 수 있다. 다른 하나는 PDDP (Principal Direction Divisive Partitioning) 클러스터링 방식이다. 이 방식은 처음에 문서들을 모으기 위해서 제일 주요한 방향(Principal Component)을 계산한 후에 하이퍼플래인(hyperplane) 결과에 따라서 두 개의 클러

스터로 나누는 방식이다. 이러한 과정을 새로 생성되는 2개의 클러스터에 반복적으로 적용하여 최종 클러스터를 구하는 방식이다.

위에서 살펴본 두 관련 연구의 웹 검색 결과의 클러스터링은 온라인으로 작업이 수행되지만 웹 에이전트를 위한 클러스터링은 배치로 작업이 수행되므로 클러스터링 속도보다는 정확도에 더 비중을 둘 필요가 있다. 따라서 본 논문에서는 초기 클러스터 생성을 위하여 Top-down 방식의 COBWEB 알고리즘을 이용하여 동적 초기클러스터 생성을 한 후 최종 클러스터 생성을 위하여 Bottom- up 클러스터링 방식으로서 Etzioni의 단어-교차 클러스터링 방식을 적용한다.

3. 통합방식 문서 클러스터링

웹 에이전트에서 통합방식 문서 클러스터링의 사용범위는 모니터 에이전트에 의해서 추출된 사용자 관심문서를 입력값으로 사용하여 유사한 내용의 문서들끼리 클러스터를 생성한 후 사용자 프로파일 생성을 위하여 웹 에이전트의 학습 시스템에 전달되는 부분에 사용된다. 이렇게 생성된 최종 클러스터는 웹 에이전트가 학습을 수행하는 단위가 되며 각 클러스터별로 사용자 프로파일의 관심영역을 표현하게 된다.

문서 클러스터링 방식을 분류하는 기준은 여러 가지가 있지만 클러스터를 생성해 나가는 순서에 있어서 Top-down 방식과 Bottom-up 방식으로 분류할 수 있다. 본 논문에서는 Top-down 방식과 Bottom-up 방식의 장점을 통합한 통합 클러스터링 방식을 제안한다. Top-down 방식으로는 개념적 클러스터링 방식인 Cobweb을 이용하며, Bottom-up 클러스터링 방식으로는 Etzioni의 교차기반 클러스터링 방식을 이용하였다. COBWEB은 속도 면에서는 빠른 클러스터링을 수행하지만 입력 문서의 순서에 종속적이며 문서 분류 후처리를 필요로 한다. Etzioni의 클러스터링은 입력 문서의 순서에 비종속적이며 정확한 클러스터링를 수행하는 반면에 많은 계산량을 필요로 해서 실행 속도면에서 취약점을 가진다. 이러한 두 가지 클러스터링 방식을 효과적으로 통합한 통합방식 문서 클러스터링은 다음과 같은 세 단계를 거쳐 수행된다.

① COBWEB을 이용하여 문서 분류 트리를 생성한다.

② 생성된 문서 분류 트리를 기반으로 제안하는 평가함수를 이용하여 동적인 초기 클러스터 생성 방식을 통한 초기 클러스터를 생성한다.

② 초기 클러스터는 Etzioni의 교차기반 클러스터링 방식을 이용하여 최종 클러스터를 생성한다.

그림 1은 통합방식 문서 클러스터링의 과정을 설명하고 있다.

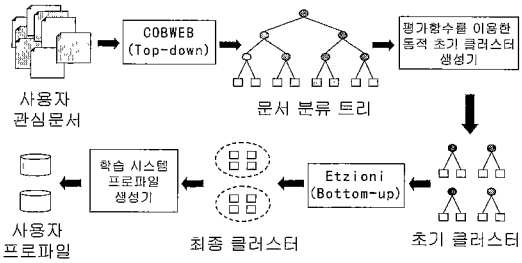


그림 1 통합방식 문서 클러스터링 과정

3.1 Top-down 클러스터링(COBWEB)

COBWEB은 원래 인간의 점진적인 개념학습을 모델링 하기 위해 개발되었다. 인간과 마찬가지로 COBWEB은 관찰을 통해서 개념을 형성해 가고 형성된 개념을 이용하여 새로운 예제를 분류할 수 있다. COBWEB은 개념적 클러스터링 알고리즘으로 다른 기계학습과 비교해서 다음과 같은 특성을 가진다[5,6,7].

- 분류 대상이 되는 개념을 계층적인 구조로 구성한다.
- 하향식 분류(Top-down Classification)를 수행한다.
- 비감독 학습(Unsupaervised Learning)방식이다.
- 점진적인 학습(Incremental Learning)을 수행한다.
- 새로운 학습대상에 대하여 힐 클라이밍(Hill Climbing) 기법을 사용한다.

COBWEB은 새로운 문서를 입력으로 받아들이게 되면 문서 분류 트리의 각 레벨에서 어떤 학습 연산자를 적용할 것인지를 평가함수의 값에 따라서 적용한다. COBWEB의 학습 연산자는 하위 노드분류(Incorporate), 새로운 노드분류(Create-new-disjunct), 병합(Merge), 분할(Split)의 4가지 연산자로 이루어져 있다. 이를 위하여 분류 트리의 각 노드에는 평가함수의 값을 계산하기 위한 노드의 상태를 저장하고 있다. COBWEB은 새로운 입력 문서가 들어올 때마다 4가지 학습 연산자를 차례로 적용하여 가장 높은 평가함수 값을 나타내는 연산자를 적용한다. 하위노드분류는 새로운 입력 문서를 현재 노드의 자식 노드에 포함시키는 연산자이다. 새로운 노드분류는 새로운 입력 문서가 기존의 노드로 분류되기에는 너무 상이한 문서인 경우에 적용하는 연산자이다. 병합 연산자는 현재 레벨의 자식 노드들 중에서 유사한 2개의 노드를 새로운 노드의 자식 노드로 변환하

는 연산자이다. 분할 연산자는 현재 노드에 속하는 자식 노드의 성격이 너무 일반화된 경우에 세분화하기 위해서 적용하는 연산자이다.

COBWEB에서 사용하는 평가함수는 카테고리 유틸리티(Category Utility)[8]를 사용한다. 카테고리 유틸리티의 기본 개념은 주어진 분류에 대하여 어떤 개체의 속성이 기대되는 기대치의 합(X)에서 분류를 고려하지 않은 기대치(Y)를 뺀 값을 주어진 분류의 개수(K)로 나눈 값으로 표현되며, 이를 수식으로 표현한 것은 다음과 같다.

$$\frac{\sum_{k=1}^{K} P(C_k)\frac{\sum_{i=1}^{I}\frac{1}{\sigma_{ik}}}{I} - \frac{\sum_{i=1}^{I}\frac{1}{\sigma_{ip}}}{I}}{4K\sqrt{\pi}}$$

위의 수식에서 K는 클래스의 개수, I는 속성의 개수, $P(C_k)$는 k 클래스로 분류될 확률, σ_{ik}는 클래스 k에서 i번째 속성의 표준편차, σ_{ip}는 부모노드에서 i번째 속성의 표준편차 값을 의미한다. 위의 수식에서 만일 σ가 0인 경우에는 $1/\sigma$의 값이 무한대가 되어 수식이 성립되지 않으므로 각 속성값의 차이를 계산할 수 있는 최소 초기치를 상수로서 정의해야 한다. COBWEB의 평가함수는 새로운 입력 문서를 기존의 분류 트리에 적용하여 계산하게 된다. 이 평가함수가 비교하는 입력 문서는 분류 트리에서 공통적인 상위 노드를 가지므로 여기서는 오직 값의 비교에 의미를 두면, 비교의 대상이 되는 각 조건들의 공통 부분을 상수라고 가정하고 위의 식을 생략한 단계의 다음과 같은 수식을 사용하여 해를 구한다[9].

$$\frac{\sum_{k=1}^{K} P(C_k)\sum_{i=1}^{I}\frac{1}{\sigma_{ik}} - \sum_{i=1}^{I}\frac{1}{\sigma_{ip}}}{K}$$

그림 2는 위에서와 같은 평가함수를 이용하여 생성된 분류 트리에서 각 노드에 저장되는 상태 정보를 설명하고 있다. 예를 들어 노드 N_3는 다음과 같은 상태 정보를 포함한다. 먼저 속성정보는 전체 문서에 속해있는 모든 속성들을 표시하며, 평균은 이들 속성들 중 노드 N_3에 속해있는 속성값들의 평균값을 나타낸다.

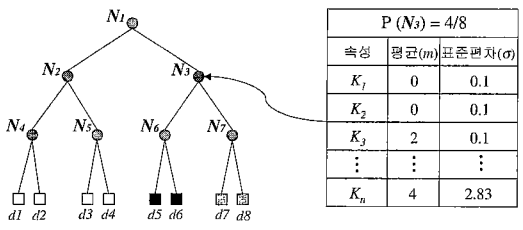


그림 2 분류 트리 노드의 상태정보 표현

다음으로 표준편차는 현재 노드 N_3에서 하위 노드에 대한 각 속성들의 표준편차를 표시하고 있다. 마지막으로 P ( )는 현재노드 N_3에 대한 확률값을 나타내고 있다.

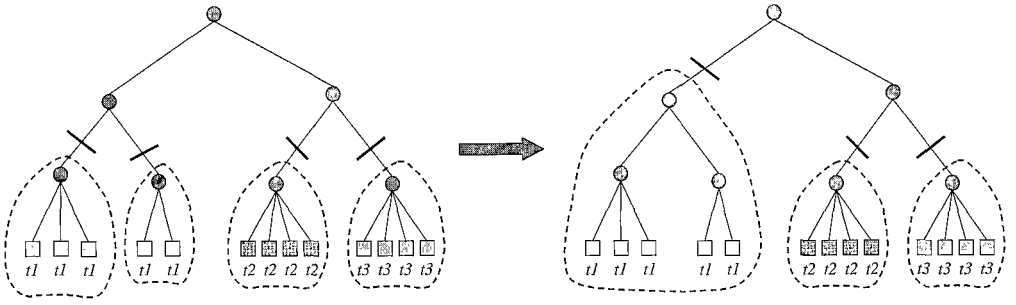
3.2 초기 클러스터 생성을 위한 평가함수

COBWEB이 생성한 문서 분류 트리는 문서들 간의 유사도를 표현하는 구조이며, 이 분류 트리로부터 초기 클러스터를 생성할 필요가 있다. COBWEB이 생성한 문서 분류 트리로부터 최종 클러스터를 바로 생성할 수 없는 이유는 COBWEB 알고리즘 자체가 입력 순서에 종속적인 성격을 가지므로 특정 노드에 다른 부류의 문서들이 포함될 수 있기 때문이다. 본 논문에서 사용한 초기 클러스터 생성 방법은 평가함수를 이용하여 분류 트리의 임의의 노드에서 초기 클러스터 생성 임계값을 만족하면 동적으로 초기 클러스터를 생성하는 방법을 사용하였다. 결과적으로 초기 클러스터의 수를 줄이게 되어, 클러스터간 병합 연산을 줄일 수 있고 클러스터링 속도면에서 향상을 가져올 수 있었다.

COBWEB이 생성한 분류 트리의 단말 노드에는 각 문서들이 할당되고 중간 노드에는 하위 노드들의 유사도를 측정하기 위한 속성 값들을 저장한다. 분류 트리로부터 초기 클러스터를 생성하기 위해서는 중간노드들에 저장되어 있는 속성 값을 이용하여 하위 노드들이 초기 클러스터를 생성하기에 충분한가를 결정하여야 한다. 다음은 본 논문에서 제안하는 초기 클러스터 생성을 위한 평가함수이다.

$$\frac{S}{D} \times \frac{n}{N} < \alpha$$

위의 수식에서 S는 현재 노드에 속하는 문서들의 유사도를 표현하는 것으로 클래스 k에서 속성 값들의 표준편차($\sum \sigma_{ik}$)의 합으로서 대입된다. D는 현재 노드에 속하는 문서들이 같은 레벨에 있는 노드에 속하는 문서들과 얼마나 유사하지 않는지를 표현하는 것으로 현재 노드의 부모 노드에서 속성 값들의 표준편차($\sum \sigma_{ip}$)의 합으로써 대입된다. 다음으로 N은 전체 입력 문서들의 개수를 의미하고 n은 현재 노드에 속하는 문서들의 개수를 의미한다. 분류 트리의 하위노드에 속하는 문서들일수록 응집도가 높아지므로 n/N 값은 응집도가 높을수록 작은 값을 가지게 된다. α는 실험에 의해 결정되는 초기 클러스터 생성을 위한 임계값이다. 각 노드에서의 표준편차의 합이 크다는 것은 하위노드들의 유사도가 낮다는 것을 의미한다. 분자인 S항은 하위노드의 문서들이 유사할수록 작은 값을 가진다. 반면에 분모인 D항은 현재 노드인 k 클래스가 형제 노드들과 다른 성격을 가질수록 큰 값을 가진다. 그러므로 현재 노드에서 보았



(a) 정적인 초기 클러스터 생성 방식 (b) 동적인 초기 클러스터 생성 방식

그림 3 초기 클러스터 생성 방식

을 때 하위노드들의 유사도가 높고 형제 노드들과 다른 점이 많은 경우에 그리고 현재 노드에 속하는 문서의 개수가 작을수록 전체적인 평가함수의 값을 감소시킨다. 위의 트리 분석 평가함수를 이용하여 분류 트리의 상위 노드에서 하위노드로 이동하며 평가하여 임계값 이하의 평가치를 보이는 노드는 하위 노드들을 병합하여 초기 클러스터로 생성할 수 있다. 그림 3은 정적인 초기 클러스터 생성방식과 동적인 초기 클러스터 생성방식을 설명한 그림이다.

(a)의 정적 방식은 COBWEB이 생성한 분류 트리에서 초기 클러스터를 생성하는 방법의 일종으로 평가함수를 적용하지 않고 정적으로(Static) 초기 클러스터를 생성하는 방식이다. COBWEB이 생성한 문서 분류 트리의 터미널 노드 위에서 각 클러스터들을 절단함으로써 초기 클러스터를 생성한다. (b)의 동적 방식은 초기 클러스터 생성을 위해서 평가함수를 이용하며 분류 트리의 임의의 노드에서 초기 클러스터 생성 임계값을 만족하면 동적으로 초기 클러스터를 생성한다. 결과적으로 초기 클러스터의 수를 줄이게 되어, 클러스터간 병합 연산을 줄일 수 있고 클러스터링 속도면에서 향상을 가져올 수 있다.

3.3 Bottom-up 클러스터링(Etzioni)

Etzioni는 웹 검색 결과를 클러스터링함으로써 사용자가 검색 결과를 쉽게 브라우징할 수 있도록 새로운 클러스터링 방식을 제안하였다. Etzioni가 제안한 클러스터링 방식은 교차기반 클러스터링 방식으로 크게 두 가지 부류로 나눌 수 있는데, 하나는 GQF라는 평가함수를 이용하는 교차기반 클러스터링 방식이고 다른 하나는 Suffix 트리를 이용하는 절-교차 클러스터링 방식이다[10]. GQF함수를 이용하는 방식은 정확도 면에서는 Suffix 트리를 이용하는 방식보다 더욱 우수하지만 클러스터링을 위한 계산량의 증가로 인하여 실행 속도 면에서는 취약점을 가진다. 하지만 사용자 프로파일을 추출하기 위한 문서 클러스터링에서는 실시간 클러스터링의 의미가 감소하기 때문에 본 논문에서 제안하는 통합 방식 문서 클러스터링에서는 GQF 함수를 사용하는 Etzioni 클러스터링 방식을 적용하되, COBWEB이 생성한 분류 트리에서 초기 클러스터를 생성한 다음 단계로 Etzioni 방식을 적용하였다.

Etzioni의 클러스터링 방식은 전통적인 HAC(Hierarchical Agglomerative Clustering) 계열로서 클러스터의 질을 정량화하기 위해서 GQF이라는 평가함수를 제안하고 있다[8]. Etzioni의 클러스터링 방식은 매번 반복할 때마다 GQF 함수의 값을 최대로 설정하는 두 개의 클러스터를 병합한다. 이런 과정은 GQF 함수의 값이 더 이상 증가하지 않을 때까지 반복한다. 이 과정을 의사코드로 표현하면 다음과 같다.

```
Initialize all documents as singleton cluster.
Until(GQF cannot be increased) do {
  Find two clusters whose merge increase GQF the most.
  Merge them.
}
```

Etzioni 방식은 클러스터간 유사도를 비교하기 위해서 모든 클러스터의 쌍을 계산해야함으로 계산 복잡도가 $O(n^2)$이 된다. 따라서 입력 문서의 개수가 증가할수록 클러스터링 수행 속도는 급속히 떨어진다. 그러나 정확도 면에서는 모든 클러스터 쌍의 유사도를 계산함으로 보다 정확한 결과값을 얻을 수 있다. 이 방식에서는 클러스터 C의 응집도를 클러스터에 속하는 모든 문서들에

공통적으로 나타나는 단어의 수로 정의하고 $h(c)$로 표시한다. 그리고 단일 클러스터의 스코어는 클러스터를 구성하는 문서의 수와 정규화된 클러스터 응집도의 곱으로 정의하고 $s(c)$로 표시한다. 아래의 수식은 클러스터 스코어를 계산하는 수식이다.

$$s(c) = |d| \times \frac{1 - e^{-\beta h(c)}}{1 + e^{-\beta h(c)}}$$

위의 수식에서 β는 클러스터의 크기와 클러스터의 응집도 사이의 Trade-off를 정하기 위한 변수이다. β 값은 경험적으로 0에서 1사이의 값을 가지는 것이 바람직하다. 그리고 문서의 개수가 한 개인 클러스터의 스코어는 0으로 정의한다. 전체 클러스터 C에 대해서 $GQF(C)$를 구하는 수식은 다음과 같다.

$$GQF(C) = \frac{f(C)}{g(|C|)} \sum_{c \in C} s(c)$$

$GQF(C)$ 함수는 크게 세 부분으로 이루어진다. 먼저 $f(C)$는 전체 문서 집합중에서 문서 개수가 두 개 이상인 클러스터에 포함되는 문서의 비율이다. 다음으로 $g(|C|)$는 클러스터의 개수를 표시한다. 그리고 $s(c)$의 합은 모든 클러스터 스코어의 합을 나타낸다. GQF 평가함수의 $1/g(|C|)$ 항과 $\sum_{c \in C} s(c)$ 항은 낮은 응집도의 소수의 클러스터를 생성하려는 경향과 높은 응집도의 다수의 클러스터를 생성하려는 경향을 각각 생성하고 GQF 함수는 적절한 Trade-off 지점을 찾아준다.

전통적인 HAC 알고리즘은 클러스터링 수행을 정지하는 임의의 임계값을 가졌던 반면에 GQF 함수는 자체 수식 내에서 클러스터 생성을 정지할 시점을 찾아준다. 또한 클러스터의 응집도를 클러스터 내의 모든 문서에 출현하는 키워드의 숫자로 정의함으로써 클러스터링 결과가 관련이 적은 문서들이 포함되는 것을 방지할 수 있다. 특히 이러한 GQF 함수의 성질은 잡음 문서들이 많이 포함될 수 있는 웹 도메인에서 좋은 클러스터링 결과를 생성할 수 있다.

4. 실험 및 결과

본 논문에서 제안하는 통합방식 문서 클러스터링의 성능을 평가하기 위해 실험에 사용될 입력 문서는 실제 웹 상에 존재하는 HTML 형식으로 이루어진 웹 페이지 원본 파일을 분야별로 수집하여 사용하였다. 웹 문서에 대한 실제 분야의 판단은 수집 단계에서 분야별 검색을 지원하는 야후(http://www.yahoo.com)와 네이버(http://www.naver.com) 검색엔진을 사용하여 각 분야별 웹 문서들을 수집하는 것으로 대체하였다. 클러스터링 방식의 비교 대상은 Etzioni 클러스터링 방식과 평가함수를 적용하지 않은 정적 클러스터링 방식을 선정하였다. 각 실험에 사용된 입력문서 집합의 구성은 다음 [표 1]과 같다.

표 1 실험의 입력 카테고리

실험	입력 카테고리
1	NBA(15), Machine Learning(15), California(15)
2	Golf(15), Music(15), Applet(15)
3	Compiler(15), Sports(15), Music(15)
4	Java(15), Parallel Processing(15), Software Agent(15)
5	Compiler(15), Golf(15), NBA(15)
6	결혼준비(13), 스타 크레프트(11), 정신건강(14)
7	Y2K(15), Chemical Weapons(15), California(15)

각 클러스터링 방식별로 17개의 카테고리에 대해서 7회 실험을 실시하였다. 각 실험 별로 3개의 카테고리를 입력으로 하였으며, 한 카테고리는 약 15개의 문서들로 이루어졌다. 실험의 성능 평가를 위한 측도로는 일반적으로 정보 검색 분야에서 많이 사용되고 있는 평가 측도들을 이용하여 각 클러스터링 방식의 성능을 비교하였다. 비교 실험을 위하여 Sun JDK1.2.1을 이용하여 각 클러스터 방식을 구현하고 PC(PentiumⅡ350) 플랫폼과 Windows NT 4.0 운영체제 환경에서 실험하였다. 다음은 본 논문에서는 제안하는 클러스터링 방식과 기존의 클러스터링 방식의 성능 비교를 위하여 사용한 성능 평가 측도(Measure)이다.

평균 정확도 ($\frac{\sum E_i(\sum C_j(|d|)/T)}{N}$)는 각 실험에서의 정확도의 평균으로, 정확도 계산은 전체 문서 중에서 최종 클러스터에 맞게 분류된 문서들의 비율로서 계산하며, 평균 정확도의 값이 높은 클러스터링 방식이 성능 면에서 더욱 우수하다고 할 수 있다. 위의 수식에서 C_j는 최종 클러스터들을 의미하며, d는 맞게 분류된 문서, T는 전체 문서의 개수를 의미한다. 그리고 E_i는 각 실험을 의미하고, N은 전체 실험 횟수를 나타낸다. 평균 클러스터 개수 오차 ($\frac{\sum E_i(|C_O - C_I|)}{N}$)는 실험의 입력 카테고리 개수와 출력 카테고리 개수와의 차를 의미한다. 입력 카테고리 개수와 출력 카테고리 개수가 근사할수록 더욱 바람직한 클러스터링 결과이므로 평균 클러스터 개수 오차의 값이 작을수록 더욱 좋은 결과로 평

가할 수 있다. 위의 수식에서 C_O는 출력 카테고리 개수를 의미하고, C_I는 입력 카테고리 개수를 의미한다. E_i는 각 실험을 표시하며, N는 전체 실험횟수를 나타낸다.

평균 실행 속도 ($\frac{\sum E_i(|T_s - T_f|)}{N}$)는 클러스터링을 시작한 시각에서 클러스터링이 완료된 시각의 차로서 계산한다. 클러스터링 수행 시간에는 클러스터링을 수행하는 시간뿐만 아니라 인덱싱(Indexing) 작업과 클러스터링 결과를 출력하는 시간 등을 포함하고 있다. 위의 수식에서 T_s는 클러스터링을 시작한 시각이고, T_f는 클러스터링이 완료된 시각이다. E_i는 각 실험을 의미하고, N은 실험 횟수를 의미한다. 그러므로 위에서 설명한 평균 정확도는 높을수록 평균 클러스터 개수 오차는 작을수록 클러스터링의 결과는 좋다고 할 수 있습니다.

다음은 각 클러스터링 방식의 실험 결과를 평균 정확도 측면, 평균 클러스터 개수 오차 측면, 평균 실행 속도 측면에서 그래프로 표현한 것이다.

각 클러스터링 방식의 실험 결과를 평균 정확도 측면에서 그래프로 표현하면 그림 4와 같다. Etzioni 클러스터링 방식의 평균 정확도는 89%, 정적 클러스터링 방식은 89%, 통합 클러스터링 방식은 91.2%의 정확도를 보였다. 정확도 측면에서는 세 가지 클러스터링 방식이 거의 유사한 실험 결과를 나타냈다. 그림 5에서 나타난 평균 클러스터 개수 오차 측면에서는 Etzioni 클러스터링 방식의 평균 클러스터 오차는 2.14, 정적 클러스터링 방식은 3, 통합 클러스터링 방식은 0.9의 클러스터 개수 오차를 보였다. 평균 클러스터 개수 오차 측면에서는 통합 방식이 다른 방식들과 비교하여 우수한 클러스터링 결과를 생성하였다.

그림 6의 평균 실행 속도 측면에서는 Etzioni 클러스터링 방식의 평균 실행 속도는 204.023초, 정적 클러스터링 방식은 113.496초, 통합 클러스터링 방식은 29.96초의 실행 속도를 보였다. 평균 실행 속도 측면에서는 통합 방식이 Etzioni 방식과 비교해서 약 6.8배 빠르고, 정적 방식과 비교하면 약 3.8배 빠른 클러스터링을 수행하였다.

다음은 초기 클러스터 생성을 위한 평가함수의 임계값을 결정하기 위한 실험 결과이다. 초기 클러스터 생성 임계값을 0.01에서 0.2까지 변화를 주었을 때, 통합문서 클러스터링에서 평균 정확도, 평균 클러스터 개수 오차, 평균 실행 시간 측면에서 비교한 결과는 다음과 같다.

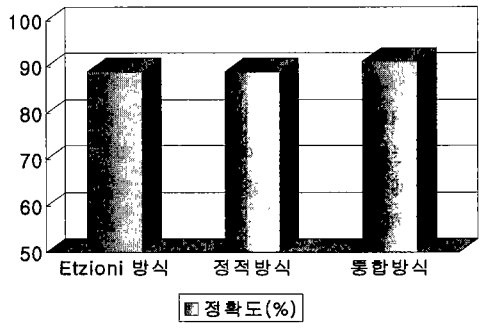


그림 4 평균 정확도 비교

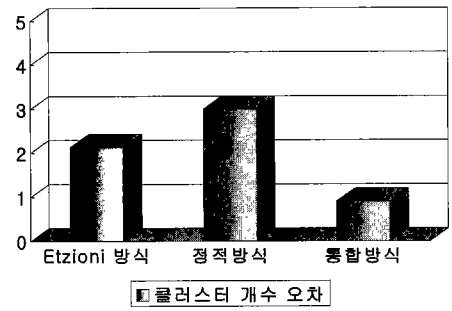


그림 5 평균 클러스터개수 오차 비교

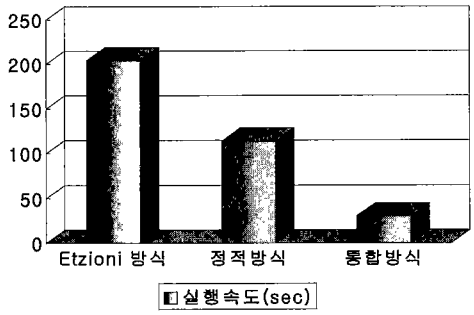


그림 6 평균 실행 속도 비교 결과

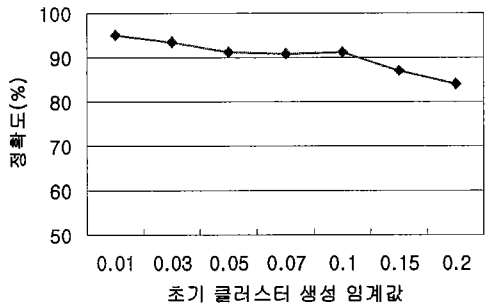


그림 7 임계값에 따른 정확도

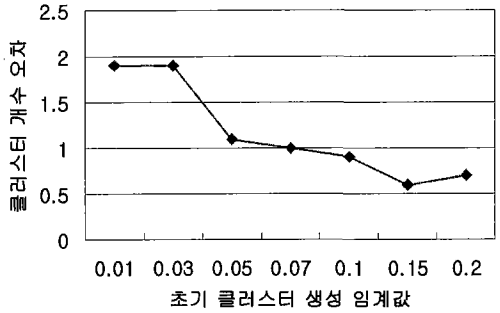


그림 8 임계값에 따른 클러스터개수 오차

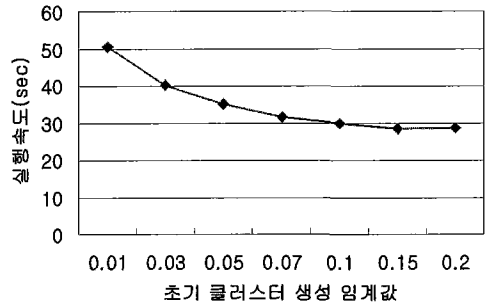


그림 9 임계값에 따른 실행 속도 비교

그림 7은 초기 클러스터 생성시 임계값의 변화에 따른 통합 클러스터링의 평균 정확도를 그래프로 표현한 것이다. 임계값이 0.01이하 일 때 최고의 클러스터링 정확도를 나타내며, 임계값이 0.1에 이르기까지는 거의 유사한 정확도를 나타낸다. 그리고 0.1 이후에는 급속히 정확도가 떨어짐을 확인할 수 있다. 그림 8은 임계값의 변화에 따른 평균 클러스터 개수 오차를 그래프로 표현한 것이다. 임계값이 0.01이하 일 때 최고로 클러스터 개수 오차가 발생하고, 임계값이 0.15에서 최저의 클러스터 오차 개수를 나타냈다. 일반적으로 임계값이 증가할수록 입력 카테고리 수와 출력 카테고리 수의 차가 줄어드는 것을 확인할 수 있다. 그림 9는 임계값의 변화에 따른 평균 실행속도 변화이다. 임계값이 0.01에서 0.1까지는 수행속도가 감소 하다가 0.1 이후로는 실행속도에 큰 변화가 없음을 알 수 있다.

실험 결과 통합 클러스터링 방식이 기존의 두 방식보다 클러스터 오차와 속도면에서 우수한 성능을 보였는데, 이와 같은 결과를 보이는 이유는 본 논문에서 제안하는 통합방식 클러스터링 방식이 Top-down 클러스터링 방식인 COBWEB의 장점과 Bottom-up 클러스터링 방식인 Etzinoi 클러스터링의 장점을 효과적으로 통합하여 정확성과 클러스터간 병합 연산의 시간을 효과적으로 줄였기 때문이다. COBWEB은 문서 분류 트리를 생성하는 속도가 $O(nlogn)$으로 선형적인 알고리즘인 반면 GQF 함수를 이용하는 Etzioni의 방식은 계산 복잡도가 $O(n^2)$이다. 그러므로 COBWEB을 이용하여 충분한 응집도와 크기를 가지는 초기 클러스터를 생성한 후에 Etzioni의 방식을 적용하면 각 문서에서부터 초기 클러스터를 생성할 때와 비교해서 클러스터간 병합 연산을 약 1/8로 줄일 수 있다. 또한 초기 클러스터 생성 임계값을 결정하기 위한 실험 결과를 분석하면 임계값이 증가할수록 평균 정확도, 클러스터 개수 오차, 실행 속도가 감소함을 확인할 수 있다. 실행속도와 정확도를 고려할 때 적정한 Trade-off 지점은 임계값이 0.1일 때이지만, 초기 클러스터에 포함된 잡음문서는 클러스터간 병합에 치명적인 악영향을 끼치므로 0.03을 초기 클러스터 생성을 위한 기본 임계값으로 설정하는 것이 바람직하다.

5. 결 론

본 논문에서는 웹 에이전트의 사용자 프로파일 생성에 도움이 될 수 있는 통합방식 문서 클러스터링을 제안하였다. 이 방식은 Top-down 방식인 COBWEB과 Bottom-up 방식인 Etzioni의 클러스터링을 효과적으로 통합하여 정확도와 속도면에서 기존의 방식보다 우수한 성능을 나타내었다. 또한 Top-down 방식과 Bottom-up 방식의 효과적인 통합을 위해서 초기 클러스터 생성을 위한 평가함수를 제안하였다. COBWEB은 입력 문서들에 대해서 카테고리 유틸리티 평가함수를 적용하여 문서 분류 트리를 생성한다. 문서 분류 트리로부터 동적으로 초기 클러스터를 생성하기 위한 평가함수를 제안하였고, 제안한 초기 클러스터 생성 평가함수를 이용하여 충분한 응집도와 크기를 만족하는 초기 클러스터를 생성할 수 있다. 따라서 최종 클러스터링된 문서 집합은 웹 에이전트가 사용자 프로파일을 학습하는데 있어서 보다 정확한 입력문서 집합을 제공하여 사용자 프로파일의 정확성을 향상시킬 수 있을 것이다.

참 고 문 헌

[1] J. Ross Quinlan, *"Induction of Decision Tree,"* Machine Learning, 1:81-106, 1986.

[2] J. Ross Quinlan, *"C4.5 Programs for Machine Learning,"* Morgan Kaufmann, San Mateo, CA, 1992.

[3] Oren Zamir, Oren Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," WWW8
[4] D. Boley, M. Gini, R. Gross, E.-H. (Sam) Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, J. Moore, *"Document Categorization and Query Generation on the World Wide Web Using WebACE"*, AI Review, 1999
[5] Doug Fisher, "Interative Optimization and Simplification of Hierarchical Clusterings," AI Access foundation and Morgan Kaufmann Publishers, 1996.
[6] Fisher, D. H., & Langley, P., "Methods of conceptual clustering and their relation to numerical taxonomy," In W. Gale(Ed.), Artificial intelligence and statistics, Reading MA: Addison Wesley, 1986.
[7] Michalski, R. S., & Stepp, R. Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), Machine learning: An artificial intelligence approach. San Mateo, CA: Morgan Kaufmann, 1983.
[8] Gluck, M., & Corter, J., "Information, uncertainty and the utility of categories," Proceedings of the Seventh Annual Conference of the Cognitive Science Society, pp. 283-287, Irvine, CA: Lawrence Erlbaum, 1985.
[9] Gennari, J. H., Langley, P., & Fisher, D. H., "Models of incremental concept formation," Artificial Intelligence, 40, pp. 11-61, 1989.
[10] Oren Zamir, Oren Etzioni, Omid Madani and Richard M. Karp, "Fast and Intuitive Clustering of Web Documents," KDD'97

양 찬 범
1998년 숭실대학교 정보과학대학 컴퓨터학부 학사. 2000년 숭실대학교 대학원 컴퓨터학과 석사. 2000년 ~ 현재 한국투자신탁 정보시스템부. 관심분야는 지능형 에이전트, 전문가 시스템, 클러스터링

이 성 열
1999년 상지대학교 전자계산학과 학사. 2001년 숭실대학교 대학원 컴퓨터학과 석사. 2000년 ~ 현재 (주)에이전트엑스피트 부설연구소. 관심분야는 웹 에이전트, 전문가 시스템, 클러스터링

박 영 택
1978년 서울대학교 전자공학과 학사. 1980년 한국과학기술원 전산학과 석사. 1992년 University of Illinois at Urbana-Champaign 전산학과 박사. 1981년 ~ 현재 숭실대학교 정보과학대학 컴퓨터학부 교수. 관심분야는 인공지능, 에이전트, 기계학습