웹 에이전트를 위한 문서 자동 분류

양 찬 범°, 박 영 택
숭실대학교 정보과학대학 컴퓨터학부

Document Autoclustering for Web Agent

Chan-Bum Yang, Young-Tack Park
(cbyang@multi.soongsil.ac.kr, park@computing.soongsil.ac.kr)

Dept. of Computer Science, Soongsil Univ.

요 약

웹 에이전트는 사용자가 웹을 브라우징하는 행위를 모니터하여 사용자의 관심정보를 학습하고 사용자가 필요로 하는 웹 상의 정보를 제공하는 시스템이다. 웹 에이전트는 사용자의 관심정보를 추출하기 위해서 귀납적 기계학습을 수행한다. 이때, 학습의 효율을 높이기 위해서는 관련이 있는 문서들을 그룹화하여 학습 시스템에 제공하여야 한다. 본 논문에서는 비감독 개념 학습 알고리즘인 COBWEB을 이용하여 사용자가 관심을 표시한 문서들의 분류 트리를 생성한다. 분류트리는 귀납적 기계학습 시스템의 입력으로 사용될 수 있는 형태가 아니므로 분류 트리의 분석과 문서 분류 후처리 작업을 통해서 문서 집합을 생성해야 한다. 이를 위해서는 분류트리를 분석하여 초기 클러스터를 생성하고, 유사한 클러스터들의 병합을 수행한다. 본 논문에서 제안하는 문서 자동 분류 방식은 비감독 개념 학습 알고리즘이 생성한 문서 분류 트리의 분석을 통해서 충분한 유사도와 적절한 수의 문서를 포함하는 초기 클러스터를 생성할 수 있다. 그러므로 문서 분류의 후처리 작업인 클러스터의 병합 작업에서 불필요한 작업을 제거함으로써 보다 효과적이고 합리적인 문서 분류 작업을 수행한다.

1. 서 론

웹상에 올라오는 문서의 수가 기하급수적으로 증가하면서 사용자가 원하는 문서를 찾기는 더욱 힘들어 지고 있다. 이러한 문제를 해결하기 위해 등장한 웹 에이전트는 사용자가 웹을 항해하는 행위를 모니터링 하여 사용자의 관심정보를 학습한다. 그리고 학습결과를 이용하여 각 사용자에게 Adaptvie한 정보를 능동적으로 제공한다.

이러한 웹 에이전트가 학습을 하기 위해서는 사용자의 관심정보를 모니터링한 결과인 웹 문서를 학습 시스템에 제공하여야 한다. 이때 학습 시스템에 제공되는 문서는 서로 관련이 있는 문서들끼리 클래스 형태로 제공이 되어야만 귀납적 기계학습 시스템의 학습 효율을 높일 수 있다. 본 논문에서는 이러한 문서 분류 작업을 수행하기 위해서 비감독 개념학습 알고리즘인 COBWEB을 이용하여 문서 분류 작업을 수행한다.

COBWEB이 생성하는 분류 트리는 각 문서들간의 유사도를 나타내는 트리로서 귀납적 기계학습 시스템의 입력으로 사용될 수 있는 형태가 아니다. 그러므로 분류 트리를 분석하여 학습 시스템의 입력에 적절한 형태로 변형하기 위해서 관련이 있는 문서들을 클래스로 생성해야 한다. 분류 트리의 후처리 작업은 2가지 작업으로 나뉘어 진다.

1단계에서는 COBWEB이 생성한 분류 트리에서 적절한 노드를 선택하여 하위 노드들을 하나의 초기 클러스터로 생성한다. 이러한 초기 클러스터를 생성하기 위해서는 분류 트리의 어느 노드를 기준으로 절단할 것인가를 정해야 한다. 본 논문에서는 분류 트리를 분석하여 하위 노드들이 초기 클러스터로 생성될 수 있는가를 결정할 수 있는 평가 방법에 대해서 설명하도록 한다.

2단계에서는 분류 트리의 분석을 통해서 생성한 초기 클러스터을 이용하여 최종 클러스터를 생성하는 작업을 수행한다. 기존의 확산/수집 (Scatter/Gather) 문서 분류 방식은 단위 문서에서부터 시작하여 클러스터를 생성해 나가는 반면에, 본 연구에서의 문서 분류 방식은 분류 트리를 분석하여 생성된 초기 클러스터가 충분한 유사도와 문서의 개수를 가지고 결합되어 있으므로 기존의 방식에서 불필요한 병합 작업을 제거하여 효율적으로 문서 분류를 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 자동 문서 분류의 관련연구를 소개하는데, 본 논문에서 이용한 분류방식인 계층문서 분류의 다른 예를 설명한다. 3장에서는 비감독 개념 학습 알고리즘인 COBWEB을 설명하고 4장에서는 COBWEB이 생성한 분류 트리의 분석을 통한 후처리를 설명한다. 5장에서는 결론 및 향후 연구에 관해 기술한다.

2. 관련 연구

문서 분류(Document Clustering)는 정보 검색 분야에서 문서를 브라우징하는 속도를 향상시키거나 검색 질의어를 확장하기 위해서 주로 이용되어 왔다. 문서 분류는 크게 계층 문서 분류와 비계층 문서 분류 방식으로 분류된다[1]. 계층 문서 분류는 일반적으로 속도가 느리다는 단점이 있지만 문서간에 유사도를 시각적으로 표현할 수 있는 장점이 있다. 이하 절에서는 계층 문서 분류 방식의 하나인 교차기반 분류(Intersection-based Clustering)방식을 설명한다.

2.1 교차기반 분류

교차기반 분류 방식은 메타검색 엔진인 메타크롤러(MetaCrawler)를 개발한 Etzioni팀에 의해 제안된 방식으로 확산/수집(Scatter/Gather) 문서 분류의 구조를 따르고 있다. 일반적인 웹 검색엔진은 검색 결과로 많은 양의 문서를 유사도 우선 순위에 따라서 제공하는데, 사용자는 많은 양의 검색 결과에서 원하는 정보를 찾기가 용이하지 않다. Etzioni 팀은 이러한 사실을 주목하고 사용자가 검색 결과를 쉽게 브라우징할 수 있도록 관련 있는 문서들을 그룹화하기 위해 교차기반 문서 분류

방식을 개발하였다[2,6].

교차기반 문서분류는 Global Quality Fuction(GQF)라는 평가함수를 이용하여 클러스터의 유사도를 평가한다. 아래의 의사코드는 단어 교차기반 문서 분류 방식의 알고리즘을 설명하고 있다.

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

교차기반 문서분류의 평가함수인 GQF는 다음과 같다.

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

위의 수식에서 $f(C)$는 전체 문서 집합에서 두 개 이상의 문서로 이루어진 집합(Cluster)을 이루는 문서의 비율을 나타내는 함수로서 집합에 포함되지 않는 분서들은 GQF값을 떨어뜨리게 된다. g(|C|)는 클러스터 수의 증가를 나타내는 함수이다. 그러므로 클러스터의 수가 증가할수록 GQF의 값은 감소한다. $s(C)$는 각 클러스터에서 공통된 키워드에 기반해서 계산된 클러스터의 응집도를 나타낸다.

Etzioni팀이 제안한 교차기반 문서 분류에서의 평가함수는 임계치를 평가함수 내에서 정규화하고 있다. 따라서 잡음 문서가 많은 검색엔진의 결과문서 분류에 대해서 잘 적용될 수 있다. 그러나 비감독 개념 학습의 결과인 분류 트리와 같은 시각적으로 사용자가 분석할 수 있는 분류 결과를 제공하고 있지 않다. 또한 본 논문에서 제안하는 문서 집합간의 병합은 초기 집합이 어느 정도 유사도를 가진 문서들의 집합이므로 Etzioni팀의 개별 문서에서부터 집합을 형성해 나가는 방식과 비교해서 더욱 효과적이라 할 수 있다.

3. 문서 자동 분류 알고리즘

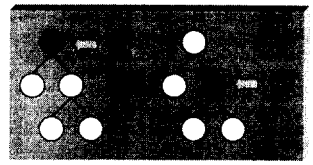
본 연구에서는 문서 자동 분류를 위해서 점진적 개념 학습 알고리즘인 COBWEB을 이용하였다. COBWEB은 원래 인간의 점진적인 개념 학습을 모델링 하기 위해 개발되었다. 인간과 마찬가지로 COBWEB은 관찰을 통해서 개념을 형성해 가고 형성된 개념을 이용하여 새로운 예제를 분류할 수 있다. 본 장에서는 사용자의 관심문서들을 입력으로 COBWEB이 어떻게 분류트리를 생성하는지를 설명한다[4,5].

3.1 점진적 개념 학습

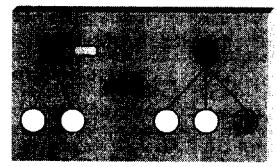
COBWEB은 점진적 개념 학습 알고리즘으로 다음과 같은 특성을 가진다. COBWEB은 학습의 결과로서 개념 계층을 형성한다. 학습의 결과인 분류트리는 하위 노드로 내려갈수록 유사한 개념을 나타내므로, 사용자에게 문서들간의 유사도를 시각적으로 전달하기에 매우 유용한 수단이다. COBWEB은 비감독 학습(Unsupervised Learning)을 수행한다. COBWEB은 입력 문서를 분류할 때, 사용자로부터 어떤 안내나 피드백없이 학습을 수행한다. 또한 COBWEB은 점진적 학습(Incremental Learning) 알고리즘으로 한번에 하나의 예제를 처리하고 이후로는 전에 입력된 예제에 대해서 재작업을 필요로 하지 않는다. 웹 에이전트에서 사용자의 관심 문서는 계속적으로 증가함으로 COBWEB의 이러한 점진적 학습 특징은 웹 에이전트의 관심문서 처리에 잘 적용될 수 있다.

3.2 COBWEB의 학습 연산자

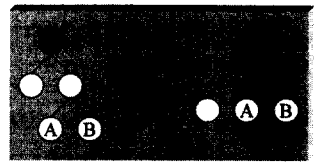
COBWEB은 새로운 문서를 입력으로 받아들이면 [그림 1]과 같은 4가지 연산자를 분류 트리에 적용을 한다. 각 연산자를 적용했을 때의 평가함수 값을 계산을 하고 가장 높은 값을 나타내는 연산자를 적용하여 새로운 문서를 분류한다. Incorporate 연산은 새로운 입력 예제를 기존의 개념에 포함을 시키고, Create New는 새로운 예제가 기존의 분류된 개념과 유사하지 않은 경우 새로운 개념을 생성한다. Merge와 Split은 새로운 입력 예제가 있을 때마다 전체적인 트리의 구조에 변화가 필요할 때 적용되는 연산이다.



Incorporate(하위노드로 분류)



Create New(새로운 노드로 분류)



Merge, Split(병합, 분할)

: New Node

: Current Node

[그림 1] COBWEB의 학습 연산자

3.3 COBWEB의 평가함수

COBWEB은 입력 예제를 속성과 값의 쌍으로서 벡터 리스트로 받아들인다. 그리고 입력 예제를 분류트리의 상위노드에서 하위노드로 분류하는데, 각 노드에서 4가지 연산자를 적용하여 가장 높은 평가함수 값을 나타내는 연산자를 실행한다. COBWEB에서 사용하는 평가함수는 Category Utility로써 클래스 멤버 내부의 원소들의 유사도는 최대로 하고 다른 클래스의 멤버와의 유사도는 최소화하는 성질을 가진다. 다음 수식은 COBWEB의 평가함수를 개념화한 것이다.

$$\frac{X - Y}{K}$$

위의 수식에서 X는 주어진 K 카테고리에 대해서 입력 예제의 속성 값이 특정 값을 가질 기대치를 의미하고 Y는 카테고리 정보가 주어지지 않은 상태에서 입력 예제의 속성 값이 특정 값을 가질 기대치를 나타낸다. COBWEB은 속성 값이 명사로 표현되는 경우에 대해서만 처리를 할 수 있으므로 아래와 같은 수식을 이용하여 속성이 수치로 표현되는 입력 예제를 처리할 수 있게 한다[3].

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

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 부모 노드에서 i번째 속성의 표준편차 값을 의미한다. 위의 수식에서 만일 σ가 0인 경우에는 $1/\sigma$의 값이 무한대가 됨으로 수식이 성립되지 않는다. 그러므로 각 속성 값의 차이를 계산할 수 있는 최소 초기치를 상수로서 정의해야 한다.

4. 문서 분류 후처리

COBWEB에 의해 생성된 분류 트리는 트리내의 문서들간의 유사도를 보여준다. 그러나 웹 에이전트의 귀납적 기계학습 시스템은 관련이 있는 문서 집합을 입력으로 받아들임으로 문서 분류를 위한 후처리 작업을 필요로 한다. 후처리 작업은 두 가지 단계로 나뉘어 진다. 첫 번째 단계에서는 COBWEB의 분류 트리를 분석하여 초기 클러스터를 생성한다. 두 번째 단계에서는 초기 클러스터간의 유사도 계산을 통해서 최종 클러스터를 생성한다. 본 장에서는 이러한 일련의 처리 과정을 기술하도록 한다.

4.1 분류 트리 분석을 통한 초기 클러스터 생성

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

$$\frac{\sum_i \sigma_{ik}}{\sum_i \sigma_{ip}} \times \beta < \alpha$$

위의 수식에서 σ_{ik}는 클래스 k에서 i번째 속성의 표준편차를 의미하고, σ_{ip}는 클래스 k의 부모 노드에서 i번째 속성의 표준편차를 의미한다. β는 계산을 용이하게 하기 위한 상수이고, α는 실험에 의해 획득된 임계값이다. 각 노드에서의 표준편차의 합이 크다는 것은 하위노드들의 유사도가 낮다는 것을 의미한다. 분자인 $\sum_i \sigma_{ik}$ 항은 하위노드의 문서들이 유사할수록 작은 값을 가진다. 반면에 분모인 $\sum_i \sigma_{ip}$ 항은 현재 노드인 k 클래스가 형제 노드들과 다른 성격을 가질수록 큰 값을 가진다. 그러므로 현재 노드에서 보았을 때 하위노드들의 유사도가 높고 형제 노드들과 다른 점이 많은 경우에 임계값인 α 보다 작은 값을 가진다. 위의 트리 분석 평가함수를 이용하여 분류 트리의 상위노드에 하위노드로 이동하며 평가하여 임계값 이하의 평가치를 보이는 노드는 하위 노드들을 합하여 초기 클러스터로 생성할 수 있다.

4.2 초기 클러스터 병합을 통한 최종 클러스터 생성

분류 트리의 분석을 통해서 생성한 초기 클러스터는 클러스터를 구성하는 문서들간의 유사도가 매우 높은 상태이다. 이와 같이 유사도의 임계값을 높인 이유는 초기 클러스터내에 잡음 문서를 포함하지 않게 하기 위해서이다. 초기 클러스터에 만일 관련이 없는 문서가 포함이 될 경우에는 클러스터간의 병합 작업에서도 계속 잡음 문서로서 남아있기 때문이다. 따라서 초기 클러스터는 세부적인 개념을 형성하는 경향이 있으므로 관련이 있는 초기 클러스터들을 병합할 필요가 있다.

클러스터들을 병합하는 작업을 수행하기 위해서는 먼저 클러스터의 대표 벡터를 표현한다. 이 대표 벡터는 클러스터의 키워드와 키워드의 발생빈도수의 리스트로 구성된다. 아래와 같은 키워드 벡터로 표현되는 두 A, B 클러스터간의 유사도는 다음의 수식으로 표현된다.

$A : \{(t_1, F_{a1}), (t_2, F_{a2}), \ldots, (t_n, F_{an})\}$
$B : \{(t_1, F_{b1}), (t_2, F_{b2}), \ldots, (t_n, F_{bn})\}$

$$\text{유사도}(A,B) = \sum_{i=1}^{n} F_{ai} \times F_{bi}$$

위의 클러스터간 유사도 계산식에 의해서 클러스터간의 유사도를 계산하고 임계값 이상의 유사도를 가지는 클러스터들을 병합한다. 예를 들어 A, B, C 3개의 클러스트간의 유사도를 고려할 때에 각 클러스터간의 유사도가 다음과 같을 경우를 가정하자. 여기서 α는 실험에 의해 얻어지는 경험적 수치인 임계값이다.

유사도(A, B) > α, 유사도(B, C) < α, 유사도(A, C) > α

3개의 클러스터가 위와 같은 유사도를 보일 때, B와 C는 임계값 이하의 유사도를 가짐에도 불구하고 A 클러스터와의 유사도 값이 임계값 이상이므로 A, B, C는 하나의 클러스터를 생성할 수 있다.

이와 같은 클러스터링 방식은 연관이 적은 클러스터를 하나의 클러스터로 생성하는 오류를 발생시킬 가능성이 있다. 또한 임계값을 너무 낮게 설정하면 병합되어야 할 클러스터들이 병합되지 못하여 단편화(Fragmentation) 현상을 나타낼 수 있다. 그러므로 임계값인 α의 설정을 위해서는 충분한 실험을 통한 경험적 수치를 얻어야 한다.

5. 결론 및 향후 연구

본 연구에서는 웹 에이전트 학습시스템의 입력으로 필요한 관심문서의 클래스를 생성하는 효과적인 클러스터링 방식을 설명하였다. 관심문서의 분류를 위해서는 점진적 개념 학습 알고리즘인 COBWEB을 이용하고, COBWEB이 생성한 문서 분류 트리의 분석을 통해서 초기 클러스터를 생성한다. 이렇게 생성된 초기 클러스터는 충분한 유사도를 가지고 세부 개념을 표현함으로 초기 클러스터의 병합을 통해서 좀 더 일반적인 개념을 형성한다. 따라서 점진적 개념 학습 알고리즘의 장점을 유지하면서 효과적으로 클러스터간의 병합을 수행할 수 있다.

한편 문서 자동분류 방식에 의해 생성된 문서 클러스터에 대해서 사용자의 피드백을 받아서 사용자의 선호도를 반영한 클러스터를 생성할 수 있게 된다면 웹 에이전트의 학습 효율을 더욱 증가시킬 수 있을 것이다. 그러므로 사용자의 피드백을 어떻게 받을 것인가와 이를 평가함수에 어떻게 반영할 것인가에 대한 연구가 필요하다.

참고 문헌

[1] E.Raumussen., *"Information Retrieval,"* pp. 419-442. Prentice Hall, Eaglewood Cliffs, N.J., 1992.

[2] Oren Zamir, Oren Etzioni, Omid Madani and Richard M. Karp, *"Fast and Intuitive Clustering of Web Documents,"* KDD'97

[3] Gennari, J. H. , Langley, P., & Fisher, D. H., *"Models of incremental cencept formation,"* Artificial Intelligence, pp. 11-61, 1989.

[4] Fisher, D. H., & Langley, P., *"Methods of conceptual clustering and their relation to numerical taxonomy,"* In W. Gale(Ed.), Artificial Intelligence and Statistics, Addison Wesley, 1986.

[5] Doug Fisher, *"Interactive Optimization and Simplification of Hierarchical Clusterings,"* AI Access Foundation and Morgan Kaufmann Publishers, 1996.

[6] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. *"Scatter/gather: a clustered-based approach to browsing large document collections,"* 15th ACM SIGIR, pp. 318-329, 1992.

[7] Moises Goldszmidt, Mehran Sahami, *"A Probabilistic Approach to Full-Text Document Clustering,"* Technical Report ITAD-433-MS-98-044, SRI International.