웹 에이전트 사용자 특성모델 구축을 위한 비감독 문서 분류

오 재 준 박 영 택
숭실대학교 정보과학대학 컴퓨터학부

Unsupervised Document Clustering for Constructing User Profile of Web Agent

Jae-Jun On, Young-Taek Park (ohjj, park@multi.soongsil.ac.kr)
Dept. of Computer Science, Soongsil Univ

요 약

본 연구는 웹 에이전트에 있어서 가장 핵심적인 부분이라 할 수 있는 사용자 특성모델 구축방법을 개선하는데 목적을 두고 있다. 사용자 특성모델을 귀납적 기계학습 방식으로 자동 추출하기 위해서는, 사용자가 관심을 가지는 분야별로 문서를 자동 분류하는 작업이 매우 중요하다. 지금까지의 방식은 사람이 관심여부에 따라 문서를 수동적으로 분류해 왔으나, 문서의 양이 기하급수적으로 증가할 경우 처리할 수 있는 문서의 양에는 한계가 있을 수밖에 없다. 또한 수작업 문서 분류 방식을 웹 에이전트에 그대로 적용하였을 경우 사용자가 일일이 문서를 분류해야한다는 번거로움으로 인해 웹 에이전트의 효용성이 반감될 것이다. 따라서 본 연구에서는 비감독 문서 분류 알고리즘과 그것을 바탕으로 얻어진 문서 분류 정보를 후처리(Post-Processing)함으로써 보다 간결하고 정확한 문서 분류 결과를 얻을 수 있는 구체적인 방법을 제공하고자 한다

1. 서 론

웹 에이전트의 기본적인 역할은 사용자가 관심을 가지는 정보의 주제를 파악하여 관련된 정보를 수집, 제공하는 것이라 할 수 있다. 웹 에이전트가 이 같은 기본적인 역할을 효과적으로 수행하여 실제 개인의 정보검색에 도움을 주기 위해서는 무엇보다 사용자가 선호하는 정보의 주제를 파악하는 일이 라 할 것이다. 즉 사용자 특성모델을 얼마나 효과적으로 구축하느냐에 따라 웹 에이전트의 성능이 좌우된다할 수 있을 것이다. 사용자 특성모델의 구축에 있어서는 두 가지 방법을 고려해 볼 수 있는데, 첫 번째, 사용자가 자신의 관심 여부를 문서별로 일일이 지정하여 특성모델 구축에 이용하는 방법으로 사용자가 직접 관심 여부를 지정하기 때문에 매우 정확한 특성모델을 구축할 수 있는 반면 문서 하나 하나에 대한 분류 작업을 사용자가 직접 수행한다는 치명적인 단점을 가진다. 두 번째, 사용자가 조회한 문서들을 대상으로 URL을 저장하거나, 문서를 별도로 저장 또는 인쇄하는등, 관심의 기진다고 판단되는 문서들에 대해 분류함수(Category Utility)을 적용하여 사용자가 관심을 가지는 분야의 주제를 파악한 후 이를 특성모델 구축에 이용하는 방법이 있다. 후자의 경우에는 전자가 가지는 단점을 해결할 수 있는 반면 어떠한 문서 분류 방법을 적용하느냐에 따라 사용자 특성모델의 정확도가 결정되므로 효과적인 문서 분류 방법의 적용은 웹 에이전트에 있어서 필수적이라 할 것이다.

근래에는 문서를 분류함에 있어서 단순히 관련이 있는 문서를 추출하는 것에 그치지 않고 문서라는 일련의 입력 정보를 트레드 각 문서의 지배적인 주제를 찾아내고, 해당 문서에 대한 개요 및 숨겨진 유사성(Hidden similarity)을 파악 사용자의 요구에 대한 다양하고 간결한 정보를 제공하는 문서기공기법(Text Mining)들에 대한 연구가 활발히 진행중이다[Intelli]. 문서가공기법은 데이터가공기법(Data Mining)의 한 분야로써 기본적으로 일련의 문서들을 처리 대상으로 하며, 연속된 입력 자료들 내의 키워드의 출현 빈도, 형태, 연관성 등을 토대로 많은 양의 정보를 보다 체계적이고 효과적으로 알아볼 수 있도록 기공하는데 목적을 두고 있다[Helena97]. 효과적인 문서기공기법은 웹 에이전트의 사용자 특성모델 구축에 필요한 문서 분류 정보를 얻는데 활용될 수 있을 것으로 기대된다.

2장에서는 문서가공기법에 관해 개괄적으로 다루고, 3장에서는 본 연구에서 사용하는 비감독 범주 효용성 알고리즘인 COBWEB의 간략한 개요와, 문서 분석에의 적용에 대한 방법론에 대하여 설명하였으며, 4장에서는 COBWEB을 이용하여 얻어진 군집 분류정보를 후처리(Post-Processing)해야할 필요성에 대해 설명하고, 5장에서는 분류트리의 후처리결과와 향후 연구방향에 대하여 기술한다.

2. 관련 연구

데이터가공기법(Data Mining)의 한 분야라 할 수 있는 문서가공기법(Text Mining)은 처리하여야할 문서의 양이 많이짐에 따라 다양한 분야에서 그 필요성이 대두되고 있다. 문서가공기법의 목적은 수집된 문서들을 대상으로 분류작업 및 문서가 내포하고 있는 정보들을 체계화하여 사용자의 요구에 따라 가공된 정보를 제공하는 데 있다. 문서가공기법에서 기본적으로 주목하고 있는 부분은 수집된 문서들을 대상으로 키워드 추출, 동시 출현 단(Co-occurring term)파악, 동의어, 파생어, 반의어 인식, 다국어 해석, 문서간 연관성 파악 등의 다양한 처리과정을 거쳐 사용자의 요구(Query)에 맞는 형태로 문서처리 결과를 제공하는 것이다[Intelli]. 흔히 사용되는 간단한 문서비교 방식이 문서 내에서 찾아낸 키워드의 빈도에 의존하고 있는 반면 문서가공기법에서 비중을 두고 있는 부분은 키워드의 출현빈도 외에 단어가 문장내에서 가지는 문맥적 의미와 언어적 특성을 분류 및 정보기공에 활용하는 데 있다 할 것이다.

2.1 IBM Intelligent Miner for Text

Intelligent Miner는 IBM에서 연구중인 문서가공기법을 이용한 툴킷(toolkit)으로써, 비정형적인 문서들을 기업체 관공서, 또는 개인이 필요로 하는 정보로 변환시켜 이용할 수 있도록 해주는 유틸리티 소프

트웨어이다.

Intelligent Miner는 문서로부터 키워드를 추출하고, 문서들을 주제별로 분류한 다음 분류된 문서들의 그룹에서 지배적인 주제를 찾아내 사용자로 하여금 필요에 따라 원하는 문서를 찾고, 찾아낸 문서로부터 다양한 정보를 얻어낼 수 있도록 해 준다. 이와 같은 절차를 자세히 살펴보면 다음과 같다. 먼저, 문서로부터 키워드를 추출해 내는 특성추출(Feature Extraction) 단계에서는 사람, 조직, 장소 등의 이름 및 동일 인명에 대한 다른 단어간의 링크 구축, 다양한 방면의 전문용어, 약어, 숫자, 날짜 처리가 포함되어 보다 융통성 있는 키워드 추출을 가능하게 한다. 다음으로, 추출된 키워드에 따라 문서들을 분류하는 단계에서는 문서집합이 갖는 개요, 숨겨진 유사성 파악(hidden similarity), 예외 인식 등으로 문서집합에 대한 정보조회를 용이하게 한다.

Intelligent Miner는 Basic Retrieval System과 Text Mining Enhancement 두 부분으로 구성되어 있는데 Basic Retrieval System에서는 16개의 언어를 지원하며, 동의어, 파생어, 반의어 등 세련된 언어학적 처리기능을 제공하고 하나의 검색 엔진으로 여러 개의 영역을 검색할 수 있도록 해 준다. Text Mining Enhancement에서는 키워드 추출과 동시에 키워드에 대한 색인을 생성하여 사용자 조회 시 사용될 수 있도록 하며, 사전에 사용자가 서로 관련된 문서와 그렇지 않은 문서들을 지정하도록 하여 정확도를 향상시킨다.

이처럼 Intelligent Miner를 비롯한 문서가공기법 분야에서 문서를 대상으로 분류 및 정보가공에 사용되는 기법들은 웹 에이전트의 특성모델 구축을 위한 문서분류에도 응용될 수 있을 것이다.

3. 비감독 개념학습

본 연구에서는 사용자 특성모델을 구축하기 위한 문서 분류 방식으로 COBWEB 알고리즘을 이용하였다. 본 장에서는 COBWEB 알고리즘과 문서로부터 추출된 키워드 벡터들을 대상으로 COBWEB 알고리즘을 적용하여 중간분류결과를 얻기 위한 세부사항에 관해 기술한다.

3.1 COBWEB

COBWEB은 점진적인 개념 형성(Incremental Concept Formation)에 기초한 학습 알고리즘으로서, 범주 정보가 주어지지 않은 예제들을 대상으로 각 예제그룹의 내용에 따라 트리의 형태로 분류정보를 도출해내는 알고리즘이다[Fisher96]. 점진적 개념 형성 모델은 분류 대상이 되는 예제집합에 새로운 예제가 추가되더라도 모든 예제들을 다시 처리하지 않고 추가된 예제들만 처리 대상에 포함된다는 장점을 갖기 때문에 웹 에이전트의 특성모델 구축에 쉽게 적용될 수 있다.

3.2 COBWEB 알고리즘

COBWEB의 입력 자료는 속성-값의 쌍으로 이루어진 하나 이상의 특성치로 주어진다. 여기서 속성은 모든 입력 자료에 공통적으로 등장하는 것이어야 한다. COBWEB에서 새로운 입력 예제를 분류트리에 추가하기 위해서는 트리의 최상위 노드부터 예제를 대상으로 분류함수(Category Utility)를 적용하여 예제를 배정할 하위노드를 결정하여 예제가 단말 노드에 배정될 때까지 이같은 과정을 반복하여 적용하게 된다. 분류함수의 기본 개념은 새로운 예제가 들어왔을 때 현 단계에서 예제를 각각의 가능한 분류경로(자식노드)로 분류하였을 때의 유사도 값에서 분류하지 않았을 때의 유사도 값을 뺀 결과값이 가장 큰 분류경로를 선택하고, 이같은 분류절차를 예제가 단말 노드에 이를때까지 반복한다는 것이다.

$$\frac{\sum_{k=1}^{K} P(C_k)\sum_{i=1}^{I}\sum_{j=1}^{J} P(A_i=V_{ij}\mid C_k)^2-\sum_{i=1}^{I}\sum_{j=1}^{J} P(A_i=V_{ij})^2}{K}$$

위 수식에서 $P(C_k)$란 분류 k(자식노드)의 전체(전체노드)에 대한 비율이고, $P(A_i = V_{ij} \mid C_k)$는 주어진 분류에 대하여 개체의 속성이 특정 값을 가지는 확률을 나타내며, I는 속성의 개수, J는 학습개체의 개수를 나타낸다.

이러한 분류함수는 COBWEB에서 하나의 학습예제가 입력됨에 따라 분류트리의 루트노드 단계에서부터 예제가 단말노드에 이를때까지 반복적용되며 예제는 분류함수의 값에 따라 새로운 노드로 분류되거나 기존의 노드로 분류될 수 있고 기존의 노드가 병합되거나 분할될 수 있다.

본 연구에서는 비교연산 자체의 효율을 높이기 위해 위의 식을 다음과 같은 수식으로 변경하여 사용한다[Gennari89].

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

위에서 설명한 분류함수를 바탕으로 측정된 평가 수치를 적용하여 COBWEB은 입력된 각 예제를 이전까지 구축된 분류트리에 추가시키면서, 그 내용에 따라서 계층의 내용을 변경시키는 작업까지 수행하게 된다. 이러한 작업을 COBWEB에서는 대표적인 4가지 유형으로 분류하여 각 경우에 가장 최적인 범주 측정수치를 결과로 하는 적절한 분류연산을 적용하게 된다. 분류연산의 종류는 Incorporate, Create-new-disjunct, Merge, 그리고 Split이 있으며, Incorporate의 경우, 새로 입력되는 예제를 기존의 하위 범주에 포함시키는 것이며, Create-new-disjunct는 새로 입력된 예제가 기존 분류 범주와 유사성이 적을 때 새로운 하위 범주를 생성하는 것이며, Merge와 Split은 새로 입력되는 예제의 내용에 따라 각각 기존 범주계층 구조를 병합하거나 분할하여 분류트리의 형태를 변경하는 작업이다.

새로 입력되는 모든 예제들을 대상으로 기존의 분류트리의 최상위 노드부터 분류함수를 반복 적용하면 기존의 분류트리는 모든 입력 예제들을 포함하는 새로운 형태의 분류트리로 변환된다.

4. 분류 트리의 후처리(Post-Processing)

입력 문서들을 대상으로 COBWEB 알고리즘을 수행하여 얻어지는 분류트리는 예제 문서들이 갖는 내용(키워드)에 따라 다양한 형태를 가지며 그것 자체로는 문서들의 유사성에 따른 하나의 트리에 지나지 않으므로 사용자 특성모델 구축 등에 적용하기 위해서는 분류트리를 해석하여 변환하는 과정이 필요하다. 본 연구에서는 분류트리를 이용하여 문서들을 주제별로 분류한 후 이를 사용자 특성모델 구축에 이용하는 것이 목적이므로 먼저 분류트리를 통해 문서들을 주제별로 분류하는 데 주목하고자 한다.

입력된 예제 문서들은 기본적으로 문서들 간의 유사성에 따라 트리의 형태로 분류된 것이므로 동일한 부모노드를 갖는 자식노드들에 해당되는 문서들은 비슷한 주제를 갖고 있는 것으로 간주한다. 따라서 분류트리를 구성하는 여러 개의 하위 트리를 하나의 그룹으로 인식할 수 있다. 물론 하나의 트리 내에서 하위 트리를 분류해 내는 방법은 트리가 가지는 노드의 개수에 비례하여 매우 다양하게 존재한다. 따라서 분류트리로 부터 문서들을 각각의 주제별 그룹으로 분류해 내기 위해서는 분류트리를 몇 개의 하위 트리로 분류하고, 어떤 노드를 하위트리의 최상위노드로 하느냐를 결정해야 한다. 여기서는 분류트리를 여러 개의 하위트리로 분할하는 과정을 소그룹 분할이라 하기로 한다.

분류트리의 후처리 과정에서는 위에서 언급한대로 분류트리의 소그룹 분할과와 분할된 소그룹들을 각 그룹간 유사도에 따라 병합하여 최종 분류결과를 얻어내는 처리를 하게 된다.

분류트리로부터의 소그룹 분할에 있어서 하나의 소그룹은 서로 다른 주제를 갖는 문서들이 가능한 한 포함되지 않도록 분할되어야 하므로 내부에 포함된 단말 노드의 개수는 충분히 작은 수준으로 유지되어야 하며, 소그룹의 개수가 많아질수록 그룹간 유사도 비교연산의 복잡도가 증가하므로 연산의 복잡도를 악화시키지 않도록 충분히 큰 수준으로 유지되어야 한다. 일반적으로 하나의 주제를 갖는 문서들이 여러 개의 소그룹으로 나뉘어 존재하기 때문에 서로 유사한 주제를 갖는 소그룹들은 하나의 주제를 갖는 그룹으로 통합하는 과정이 필요하다

4.1 소그룹 분할 방법

소그룹 분할이란 COBWEB 알고리즘의 결과로 얻어진 분류 트리를 몇 개의 노드로 구성된 하나의 단위로 떼어내는 것을 말한다. 분류트리로부터 소그룹을 분할해낼 때에는 트리 내의 분할조건에 맞는 노드를 선택하여 해당 노드를 최상위 노드로 하는 하위트리를 하나의 소그룹으로 인식하는 방법을 사용한다. 분할조건은 선택된 노드가 단말 노드를 포함하고 있거나 자식 노드가 2인 노드를 포함하는 것으로 하며 분할의 작업은 최상위 노드로부터 단말노드로 진행해 나간다. 이러한 분할방식을 적용하면 하나의 소그룹이 갖는 문서의 개수는 최소한 3개 이상이 되므로 소그룹간 비교연산의 복잡도를 낮출 수 있다

4.2 소그룹간 유사도 측정 및 병합

소그룹분할 알고리즘에 의해 분류 트리로 부터 얻어진 여러 개의 소그룹들은 각각에 포함된 단말 노드가 가리키는 문서들이 가지는 키워드들을 토대로 유사도 비교연산에 의해 몇 개의 대그룹으로 최종 분류되어 사용자 특성모델 구축에 이용된다. 유사도 비교 연산은 각 소그룹이 가지는 대표 키워드 벡터를 이용하는데, 소그룹 Gs1, Gs2가 각각 v1 = {K1, K2 , Kn}, v2 = { L1, L2,. Lm }의 키워드 벡터를 갖는다면 다음 공식에 의해 유사도를 비교할 수 있다

$$\sum_{i}^{n}(K_i \cdot L_i)$$

이 때 유사도 비교연산에 사용된 소그룹의 대표 벡터는 소그룹에 포함된 문서들이 가지는 키워드의 출현빈도(Term Frequency)값에 대한 정규화(Cosine normalization)처리 후 얻어지는 값이 된다. 대표 벡터 내의 원소 Ki를 정규화한 후의 값 Ki'는 다음 공식에 의해 구할 수 있다

$$K'_i = \frac{K_i}{\sqrt{\sum_{i}^{n} K_i^2}}$$

두 소그룹 간 유사도 비교에 있어서 소그룹의 개수가 n이라 할 때 비교 연산은 n*(n-1)/2 번 수행되어야 한다. 비교 연산 결과 값을 정렬한 후 일정 값 이상을 갖는 소그룹들을 대상으로 큰 값을 가지는 소그룹들을 우선적으로 병합해 나가면 된다.

5. 실험 및 향후 연구

실험에서는 웹 상에서 수집된 HTML 형태의 문서들로부터 키워드 벡터를 추출한 후 COBWEB 알고리즘을 이용하여 분류트리를 구한 다음 분류트리를 대상으로 후처리 과정을 통해 최종적으로 문서들을 분류해 내는 순으로 진행되었다. 실험은 2회로 나뉘어 진행되었으며 1차 실험에서 사용된 예제 문서들은 애플릿 20개, 컴파일러 30개, 음악 40개, NBA 30개로 모두 120개이며, 2차 실험에서는 골프 20개, 캘리포니아 20개, Y2K 20개, 자바 20개로 모두 80개를 이용하였다. 4장에서 언급한 소그룹 분할 방식을 적용하여 1차 실험에서는 COBWEB 알고리즘을 통해 얻어진 분류트리로 부터 8개의 소그룹이 분할되었고 2차 실험에서는 7개의 소그룹이 분할되었다

다음은 분할된 소그룹들을 대상으로 유사도 비교연산을 수행한후 유사도가 0.50이상을 보인 소그룹들을 병합하여 최종 분류 결과를 나타낸 표이다.

실험 1		실험 2	
애플릿 20개, 컴파일러 30개, 음악 40개, NBA 30개		골프 20개, 캘리포니아 20개, Y2K 20개, 자바 20개	
그룹 1	컴파일러(28),음악(1) NBA(1)	그룹 1	골프(20)
그룹 2	NBA(29)	그룹 2	Y2K(19)
그룹 3	애플릿(20),컴파일러(2)	그룹 3	자바(20)
그룹 4	음악(39)	그룹 4	캘리포니아(20),Y2K(1)

본 연구에서는 단순히 문서가 포함하는 키워드의 출현 빈도만을 토대로 문서간 유사도 비교 연산을 수행하였다. 동일한 주제를 갖는 문서라 하더라도 어떠한 방향에서 주제를 다루고있는지에 따라 키워드의 출현 빈도에 의한 분류 알고리즘 수행결과가 전혀 예상치 못한 것이 될 수 있다. 이러한 문제는 문서간 유사도 비교에 있어서 문서가공기법에서 다루고 있는 문서 내 숨겨진 유사성 파악, 동시 출현 단어(Co-occurrence term)인식, 관련이 있는 단어들의 그룹은 짧은 간격을 두고 여러 번 반복된다는 어의적 유사성(lexical affinity) 처리등이 결여되어 있기 때문에 발생할 수 있다[Intelli]. 따라서 보다 정확한 문서 분류 결과를 기대하기 위해서는 문서가공기법에서 주목하고 있는 이상의 여러 가지 문제에 대한 연구가 진행되어야할 것이다

참고 문헌

[Helena97] Helena Ahnon & Oskari Heinonen, Mining in the Phrasal Frontier, University of Helsinki, Department of Computer Science, 1997

[Intelli] IBM Intelligent Miner for Text, http //www software ibm com

[Fisher86] 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

[Fisher96] Doug Fisher, *Interative Optimization and Simplification of Hierarchical Clusterings*, AI Access foundation and Morgan Kaufmann Publishers, 1996.

[Gennari89] Gennari, J. H , Langley, P , & Fisher D H., *Models of incremental concept formation*, Artificial Intelligence, 40, pp 11-61, 1989

[Gluck85] 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

[Joachims96] Thorsten Joachims, A *Probabilistic Anylysis of the Rocchio Algorithm with TFIDF for Text Categorization*, March 1996