북 마크 자동 분류를 위한 학습 에이전트

김 인 철† · 조 수 선††

요 약

웹은 이제 인터넷의 중요한 서비스중의 하나가 되었다. 웹 공간을 탐색할 때 사용자들은 항해하는 동안 만나는 흥미 있는 사이트들을 기록하기 위해 북 마크 기능을 이용한다. 북 마크 기능을 이용할 때 겪는 문제중의 하나가 거듭된 새로운 북 마크의 추가로 인해 북 마크 리스트의 길이가 길어지면 북 마크 리스트가 일관성 있는 구성을 잃어버리게 되어 실제적인 도움을 주기 어렵다는 것이다. 사용자가 북 마크 파일을 효율적이고 체계적으로 유지하기 위해서는 북 마크 파일에 추가되는 새로운 북 마크들을 카테고리별로 분류하여 신규 폴더를 만들거나 기존의 폴더를 찾아 삽입해주어야 한다. 본 논문에서는 대응되는 웹 문서들을 다운 받아 내용을 분석함으로써 자동으로 북 마크를 분류하는 BClassifier라 불리는 학습 에이전트를 소개한다. BClassifier 에이전트를 위한 훈련 예의 주된 공급원은 바로 사용자가 명시적으로 이미 주제에 따라 몇 개의 북 마크 폴더들로 분류해놓은 북 마크들이다. 여기에 주제 카테고리들을 확대하고 이들에 대한 훈련 문서들을 확보하기 위해 추가적으로 Yahoo 사이트의 최상위 카테고리들로부터 웹 문서들을 수집하여 훈련 예에 포함시킨다. BClassifier 에이전트는 잘 알려진 확률기반의 분류 기술인 나이브 베이지안 학습 방법을 채용하고 있다. 본 논문에서는 BClassifier 에이전트에 관한 몇 가지 실험 결과를 소개하고 평가한다. 나이브 베이지안 방법과 k-최근접 이웃 방법, TFIDF 등과 같은 서로 다른 학습 방법들과의 비교실험 결과도 제시한다.

A Learning Agent for Automatic Bookmark Classification

In Cheol Kim† · Soo Sun Cho††

ABSTRACT

The World Wide Web has become one of the major services provided through Internet. When searching the vast web space, users use bookmarking facilities to record the sites of interests encountered during the course of navigation. One of the typical problems arising from bookmarking is that the list of bookmarks lose coherent organization when the list becomes too lengthy, thus ceasing to function as a practical finding aid. In order to maintain the bookmark file in an efficient, organized manner, the user has to classify all the bookmarks newly added to the file, and update the folders. This paper introduces our learning agent called BClassifier that automatically classifies bookmarks by analyzing the contents of the corresponding web documents. The chief source for the training examples are the bookmarks already classified into several bookmark folders according to their subject by the user. Additionally, the web pages found under top categories of Yahoo site are collected and included in the training examples for the purpose of diversifying the subject categories to be represented, and the training examples for these categories as well. Our agent employs naive Bayesian learning method that is a well-tested, probability-based categorizing technique. In this paper, the outcome of some experimentation is also outlined and evaluated. A comparison of naive Bayesian learning method alongside other learning methods such as k-Nearest Neighbor and TFIDF is also presented.

키워드 : 에이전트(Agent), 웹(Web), 기계학습(Machine Learning), 북 마크 분류(Bookmark Classification)

1. 서 론

인터넷은 정보의 보고라 불리울 만큼 수 많은 정보 자원이 곳곳에 산재되어 있으며 하루에도 수 많은 정보들이 새로 부가 되고 있다. 원하는 정보를 찾아 드넓은 웹 공간을 항해하는 사용자들에게는 이미 방문한 적이 있는 유용한 웹 사이트들을 기록해두고 관리하는 일이 새로운 웹 사이트를 찾는 일만큼이나 중요하다. 그래서 넷스케이프 네비게이터와 같은 대부분의 웹 브라우저에서는 사용자로 하여금 특정 웹 사이트나 웹 문서의 주소를 기록해 두었다가 해당 웹 사이트를 재 방문할 수 있도록 북 마크 기능을 제공하고 있다. 그러나 북 마크를 효율적으로 사용하기 위해서는 늘어나는 북 마크들을 주제별로 분류하고, 동일 주제나 유사 주제별로 북 마크들을 모아 재정렬 하는 체계적인 관리작업이 필요하다. 하지만 현재 이러한 북 마크 관리 작업에는 다음과 같은 몇 가지 어려움이 있다. 먼저 북 마크 증가에 따라 북 마크 관리 작업이 일회성이 아니라 지속적으로 이루어져야 한다는 점이다. 둘째는 이러한 북 마크 관리 작업이 대부분 수작업으로 이루어져야 한다는 점이다. 현재 대부분의 웹 브라우저에

† 종신회원 : 경기대학교 정보과학부 전자계산학전공 교수
†† 정 회 원 : 한국전자통신연구원 컴퓨터 소프트웨어연구소 선임연구원
논문접수 : 2001년 8월 23일, 심사완료 : 2001년 9월 26일

서는 북 마크 관리를 위한 북 마크 편집기가 있으나, 내용 분석에 따른 각 북 마크의 분류 작업과 주제별로 북 마크들을 그룹화해주는 재정렬 작업의 대부분을 결국 사용자의 수작업에 의존하고 있다. 이러한 문제의 한가지 대안으로서, 본 논문에서는 사용자가 기록하는 북 마크들을 주제별로 자동 분류하여 주제당 하나의 북 마크 폴더 단위로 재정렬 해주는 북 마크 자동 분류 에이전트인 BClassifier를 설계, 구현하였다. BClassifier는 URL로만 구성된 북 마크들을 직접 분류하기 보다는 북 마크가 가리키는 인터넷 웹 문서에 대해 문서 분류 기계 학습법[1]을 적용함으로써 해당 북 마크를 분류하게 된다. 이 에이전트에서는 문서 분류 기계학습 방법 중에서 대표적인 나이브 베이지안 학습 방법을 이용하는데, 이 학습 방법은 확률기반의 교사학습(supervised learning) 방법으로서 다수의 훈련 예(training example)를 이용하여 학습이 이루어진다. BClassifier에서 훈련 예는 일차적으로 사용자가 이미 주제에 따라 별도의 북 마크 폴더로 분류해 놓은 북 마크와 해당 웹 문서를 사용하며, 사용자가 직접 기술해 주지 않은 카테고리(category)와 훈련 예를 확보하기 위하여 Yahoo 사이트의 최상위 분류체계와 해당 웹 문서를 보조적으로 사용한다. 한편 본 논문의 마지막 부분에서는 실험을 통하여 BClassifier 에이전트의 전체적인 성능 분석과 나이브 베이지안, k-NN, TFIDF 등의 서로 다른 3가지 학습기법에 대한 성능 비교를 시행하였다.

2. 분류 학습법과 분류 에이전트

대표적인 문서 분류 학습법에는 나이브 베이지안 기법과 k-NN기법, TFIDF기법 등이 있다[2, 3]. 이 절에서는 이들 분류 학습법과 대표적인 분류 에이전트에 대해 살펴본다.

2.1 분류 학습법

2.1.1 나이브 베이지안

나이브 베이지안(naive Bayesian) 학습기법[1, 4]은 베이스 정리(Bayes theorem)에 기초한 확률 모델을 이용한다. 이 방법에서는 분류하고자 하는 문서 d에 대한 벡터모델($w_1, w_2, \cdots, w_n$)을 입력하여, 분류 가능한 클래스 - 본 논문에서는 카테고리(category)와 클래스(class)를 혼용하여 사용함 - 들 가운데 이 문서를 관찰할 수 있는 가능성이 가장 높은 클래스를 찾아 그 클래스로 분류한다. 즉, 아래의 식 (1)과 같이 문서 d에 대한 조건부 확률이 가장 큰 클래스로 분류한다[5].

$$\arg\max_{c \in C} P(c \mid d) = \arg\max_{c \in C} P(c \mid w_1, w_2, , \cdots, w_n) \quad (1)$$

$$= \arg\max_{c \in C} \frac{P(w_1, w_2, \cdots, w_n \mid c) P(c)}{P(w_1, w_2, \cdots, w_n)}$$

$$= \arg\max_{c \in C} P(w_1, w_2, \cdots, w_n \mid c) P(c)$$

여기서 확률 $P(w_1, w_2, \cdots, w_n)$는 하나의 상수(constant)인 정규화 항(normalizing term)이므로 우리가 가장 가능성이 높은 하나의 클래스를 결정하는 것에만 관심이 있는 경우 생략 가능하다. 또한 이 학습기법에서는 식 (2)과 같이 한 문서를 나타내는 특성(feature)들인 각 w_i들 간에는 서로 조건부 독립(conditionally independent)이라는 나이브 베이지안 가정(naive Bayesian assumption)을 적용한다[1].

$$P(w_1, \cdots, w_n \mid c) = \prod_{i=1, n} P(w_i \mid c) \quad (2)$$

따라서 결론적으로 나이브 베이지안 분류 학습기법은 분류 대상 문서 d에 대해 가장 가능성이 높은 분류 클래스를 식 (3)과 같이 계산한다.

$$\arg\max_{c \in C} P(c) \prod_{i=1, n} P(w_i \mid c) \quad (3)$$

2.1.2 k-NN

대표적인 또 다른 문서 분류 학습기법으로는 최근접 이웃 방법인 k-NN(k-Nearest Neighbor)학습기법[1]이 있다. 이 방법은 식 (4)와 같이 분류대상 문서 $d = (w_1, \cdots, w_n)$와 저장되어 있는 각 훈련 문서(training document) $d' = (w_1', \cdots, w_n')$과의 유클리드 거리(Euclidian distance)를 계산하여 분류대상 문서와 가장 거리가 가까운 훈련 문서 k개를 선정한다.

$$Dist(d, d') = \sum_{i=1}^{n} \sqrt{(w_i - w_i')^2} \quad (4)$$

그리고 선정된 k개 중에서 가장 많은 훈련 문서들이 소속된 클래스로 분류대상 문서 d를 분류한다. k값은 k-NN기법의 성능을 최적화하기 위하여 일반적으로 교차검증(Cross Validation) 기법을 사용하여 사전에 결정하며, k = 1인 경우를 NN 기법이라고 한다.

2.1.3 TFIDF

전통적으로 정보검색 분야에서 많이 이용되어온 TFIDF 분류 학습기법[6]에서는 각 문서 d를 특성단어(feature word)의 출현 빈도수(frequency)에 기초한 가중치 벡터(weight vector)로 표현한다. 이때 각 단어의 가중치 w_i는 식 (5)와 같이 문서 d에 나타나는 빈도수인 TF(Term Frequency)와 그 단어가 나타나는 총 문서 수에 대한 역수인 IDF(Inverse Document Frequency)의 곱으로 계산된다. 이것은 한 단어가 특정 문서에 나타나는 빈도수는 높고 다른 문서에 나타나는 빈도수가 낮을수록 다른 문서에 비해 그 문서를 잘 표현해줄 있다는 의미를 담고 있다.

$$w_i = TF_i \cdot IDF_i \quad (5)$$

문서 분류작업을 위해서는 각 클래스별로 그 클래스를 나타내는 프로토타입 벡터(prototype vector)를 구한다. 이때

각 클래스의 프로토타입 벡터 c는 그 클래스에 속한 훈련 문서들의 (TF-IDF) 가중치 벡터들의 평균(average)으로 계산한다. 일단 이처럼 각 클래스들이 프로토타입 벡터로 표현되어 있으면, 식 (6)과 같이 분류대상 문서 d의 가중치 벡터와 각 클래스 c의 프로토타입 벡터간의 유사성(similarity)을 코사인 법칙(cosine rule)을 적용하여 계산한다. 그리고 이와 같은 과정을 거쳐 가장 유사하다고 판단되는 클래스로 문서를 분류한다.

$$\arg\max_{c \in C} \cos(c, d) = \arg\max_{c \in C} c/\|c\| \bullet d/\|d\| \quad (6)$$

2.2 분류 에이전트 시스템

일반적으로 기계 학습법[1]을 사용하여 분류 작업을 자동으로 수행할 수 있는 자율적인 소프트웨어를 분류 에이전트라 한다[7, 8]. 이와 같은 분류 에이전트의 대표적인 한 예로 카네기 멜론 대학의 Personal WebWatcher[9]가 있다. 이 분류 에이전트는 웹 브라우저를 통해 사용자의 행동을 모니터링하여 사용자의 관심영역을 학습한 뒤, 브라우징하는 웹 문서내의 링크들에 대해 사용자 관심영역에 속하는 것들과 그렇지 않은 것들을 분류하여 관심있는 링크들만을 제안 해주는 시스템이다. 또한, 앤더슨 컨설팅 연구실에서 개발된 InfoFinder[10] 역시 사용자의 관심 프로파일[11]을 바탕으로 온라인 문서들에 대한 분류작업을 통해 사용자가 관심을 가질 문서들을 찾아주는 에이전트 시스템이다. 이외에도 MIT 대학에서 만든 전자우편물을 자동 분류하는 Maxims[13], 엔터테인먼트 선별 에이전트인 Ringo, 뉴스 기사 분류 에이전트인 NewT[14] 등이 모두 문서 분류기법을 이용한 대표적인 분류 에이전트 시스템이다.

3. 시스템의 설계

3.1 기본 가정

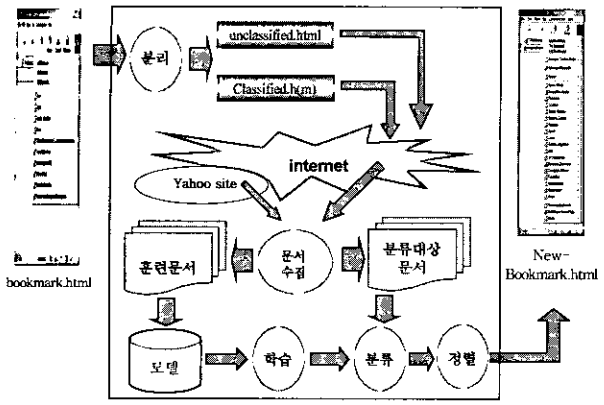
본 시스템은 효과적인 북 마크 분류를 위하여 다음과 같은 몇 가지 기본 가정을 전제로 하고 있다. 첫째, 북 마크 분류는 단지 URL과 레이블만으로 이루어진 북 마크 자체를 직접 분석하기보다는 북 마크가 가리키는 웹 문서의 내용을 대신 분석함으로써 작업이 이루어진다. 이를 위해서는 북 마크들이 가리키는 웹 문서들을 인터넷으로부터 내려 받는 일과 이러한 웹 문서들에 대해 효과적인 문서 분류 학습법을 적용하는 일, 분류된 문서에 따라 북 마크를 분류하고 정렬하는 일 등이 필요하다. 둘째, 하나의 북 마크를 분류하기 위해 대신 이용되는 웹 문서는 그 북 마크 주소가 가리키는 웹사이트(web site)의 전체 혹은 일부분의 문서들이 아니라 단 하나의 웹 문서(web document)로 제한한다. 사용자들이 북 마크를 할 때 때로는 단 하나의 특정 웹 문서를 기록해두기 위한 경우도 있고, 때로는 유용한 웹 문서들을 다수 보유하고 있는 하나의 웹 사이트를 기록해두기 위한 경우도 있다. 뿐만 아니라 비록 단 하나의 특정 문서만을 기록하기 위한 의도로 만들어진 북 마크라 하더라도 그 북 마크에 대한 효과적인 분류를 위해서는 해당 문서 외에 같은 사이트내의 이웃한 다른 문서들을 보조적으로 이용하는 것이 더 좋을 수도 있다. 하지만 본 논문에서는 북 마크를 기록할 당시의 사용자 원래 의도를 제대로 파악하기도 쉽지 않거니와 분류에 이용할 이웃한 다른 웹 문서들의 범위를 정하는 일도 쉽지 않아 이와 같은 제한을 둔다. 세째, 북 마크 분류를 위해 적용하는 문서 분류 기계 학습법은 모두 다수의 훈련 문서(training document)들을 필요로 하는 교사학습(supervised learning) 방법들이다. 본 시스템에서 사용하는 문서 분류 기법은 나이브 베이지안 기법, K-NN, TFIDF 등으로 이들은 모두 분류 클래스별로 충분한 훈련 문서들을 미리 확보하고 있어야 높은 분류성능을 기대할 수 있다. 넷째, 사용자가 직접 북 마크 편집 기능을 이용하여 주제영역에 따라 몇 개의 북 마크 폴더(bookmark folder)들을 만들고 이 폴더별로 북 마크들을 정리해두었다면 이들을 주된 훈련 예로 이용한다. 따라서 본 시스템에서는 사용자가 만든 하나의 북 마크 폴더를 사용자 관점에서 본 고유한 하나의 주제영역이자 분류 클래스로 파악하고 각 폴더 안에 위치한 북 마크들을 해당 클래스의 훈련 예로 간주한다. 다섯째, 자동 분류 작업 이전에 사용자로부터 직접적으로 충분한 분류 클래스와 훈련 예들을 얻을 수 없는 경우를 위해 대표적인 인터넷 디렉토리 서비스를 제공하는 Yahoo 사이트의 최상위 분류 클래스와 해당 문서들을 가져와 훈련 예로 이용한다.

3.2 시스템의 구조

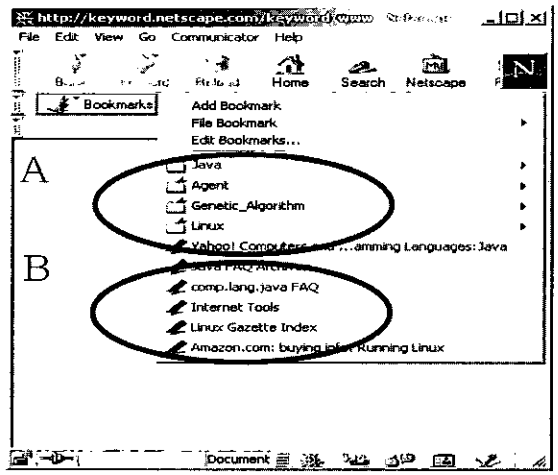
(그림 1)는 시스템의 전체적인 구조를 보여주고 있으며, 수행 과정은 다음과 같다. 먼저 웹 브라우저(web browser)의 현재 북 마크 파일(bookmark.html)에서 사용자에 의해 이미 분류된 북 마크들의 집합(classified.html)과, 분류되지 않은 북 마크들의 집합(unclassfied.html)을 분리한다. 여기서 사용자에 의해 이미 분류된 북 마크들이란 앞서 말한 바와 같이 사용자에 의해 특정 북 마크 폴더에 할당된 북 마크들을 말하며, 아직 분류되지 않은 북 마크들이란 현재 어떠한 북 마크 폴더에도 속하지 않은 북 마크들을 말한다. 따라서 이미 분류된 북 마크들에 대해서는 분류 클래스와 각 클래스에 대응하는 훈련 문서들을 확보하기 위해 인터넷 상의 웹 문서들을 수집해오고, 반면에 아직 분류되지 않은 북 마크들에 대해서는 분류대상 문서들을 확보하기 위해 북 마크들이 가리키는 인터넷상의 웹 문서들을 수집해온다. 또한 보다 충분한 분류 클래스들과 훈련 문서들을 확보하기 위해 인터넷 디렉토리 서비스를 제공하는 Yahoo 사이트로부터 최상위 14가지 클래스에 대한 웹 문서들도 수집해온다.

이와 같은 과정을 거쳐 분류작업을 위한 훈련 문서들과 분류 대상 문서들이 수집되면, 적절한 문서 전처리 과정(text/document preprocessing)과 문서 모델화(document modeling) 과정을 거친다. 그리고 이러한 분류 클래스들과 훈련 문서들을 바탕으로 사전에 문서 분류 학습법에 따라 문서 분류기를 학습하고, 이것을 바탕으로 분류 대상 문서들을 차례대로 분

류한다. 웹 문서 분류 결과에 따라 대응되는 북 마크들을 분류 클래스별로 모아서 정렬하는 작업과 이들을 기존에 이미 분류해 놓은 북 마크들과 결합하여 새로운 북 마크 파일(new-bookmark.html)을 생성하는 순서로 작업이 진행된다.



(그림 1) 시스템 전체 구조



(a)

```
<DL><p>
    <DT><H3 ADD_DATE = "949366376">[illegible]</H3>
    <DL><p>
        <DT><A HREF ="http : // java.sun.com/products/ "
ADD_DATE = "949365384" LAST_VISIT = "949365329"
LAST_MODIFIED = "949365329">Products & APIs</A>
        <DT><A
HREF = "http : // developer.java.sun.com/developer/infodocs/index.
shtml#docs" ADD_DATE = "949365399" LAST_VISIT = "949365392"
LAST_MODIFIED = "949365392">Docs & Training</A>
        <DT><A
HREF = "http : // java.sun.com/docs/faqindex.html"
ADD_DATE = "949365421" LAST_VISIT = "949365410"
LAST_MODIFIED = "949365410">Java Software FAQ Index</A>
        <DT><A HREF = "http : // www.javalobby.org/ "
ADD_DATE = "949365459" LAST_VISIT = "0"
LAST_MODIFIED = "0">Javalobby.org/</A>
        <DT><A HREF = "http : // www.javaworld.com/ "
ADD_DATE = "949365474" LAST_VISIT = "949367303"
```

(b)

(그림 2) 분류전의 북 마크 파일

3.3 북 마크 파일의 분리

(그림 2)는 에이전트에 의해 자동 분류하기 이전의 북 마크들의 모습을 보여주고 있다. 특히 (그림 2)의 (a)에서는 상용 웹 브라우저 프로그램인 네스케이프 네비게이터(Netscape Navigator)에서 북마크 폴더들과 북 마크들을 - 책갈피로도 번역됨 - 브라우징한 모습을 보여주고, 반면에 (b)는 이들에 대한 북 마크 파일의 HTML 소스코드를 보여주고 있다. (그림 2)의 (a)에서 보듯이 사용자가 이미 주제별 영역에 따라 북 마크 폴더들을 만들어 북 마크들을 분류해 놓은 것은 A 영역과 같은 모습으로 나타나고, 반대로 아직 어떤 북 마크 폴더에도 배정되지 않은 북 마크들은 B 영역과 같은 모습으로 나타난다.

(b)의 북 마크 파일 소스 코드를 보면 하나의 북 마크 폴더 안에 배정된 북 마크에 대한 HTML 태그 구성은 다음과 같다.

```
<DT><H3 ADD_DATE = "949366376"> Java</H3>
<DL><p>
<DT><A  HREF = http : //www.javalobby.org /
      ADD_DATE = "949365459"  LAST_VISIT = "0"
      LAST_MODIFIED = "O">Java Lobby</A>
</DL><p>
```

여기서 하나의 분류 클래스로 간주하는 북 마크 폴더인 "Java"와 그 폴더에 배정된 북 마크인 "Java Lobby" 부분을 살펴보면, 폴더 이름은 <DT>와 <H3> 태그로 표현하고 폴더 내의 구성원인 북 마크들에 대해서는 별도의 리스트를 구성하여 <DL> 태그로 표현한다는 것을 알 수 있다.

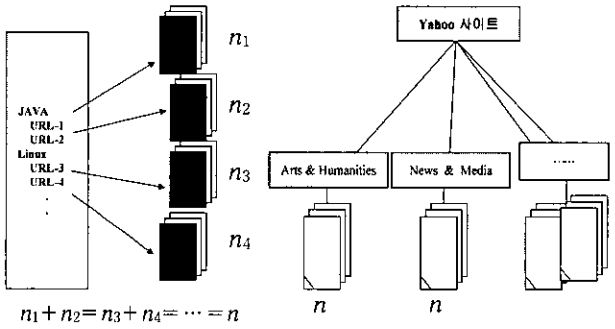
```
<DT><A  HREF = http : //www - net.com / java / faq /
      ADD_DATE = "949367554" last_visit = "949367592"
      LAST_MODIFIED = "949367523">Java FAQ Archives</A>
```

한편 특정 폴더에 배정되지 않은 채 기록된 북 마크의 하나인 "Java FAQ Archives"는 위의 소스코드에서 보듯이 단지 <DT> 태그로만 표현한다는 것을 알 수 있다. 따라서 이와 같은 HTML 소스코드 상의 차이점에 근거하여 북 마크 파일에서 이미 분류된 북 마크들(classified.html)과 분류되지 않은 북 마크들(unclassified.html)을 분리해낼 수 있다

3.4 웹 문서의 수집

북 마크 파일에 대한 분리 작업이 끝나면 본 시스템에서는 훈련 문서로 사용될 웹 문서들과 더불어 분류대상으로 사용할 웹 문서들을 인터넷으로부터 수집한다. 훈련 문서로 사용되는 웹 문서의 수집을 위해서는 classified.html에서 각 클래스 별로 북 마크된 주소들을 추출하여 인터넷 상의 해당 웹 문서를 받아서 저장한다.

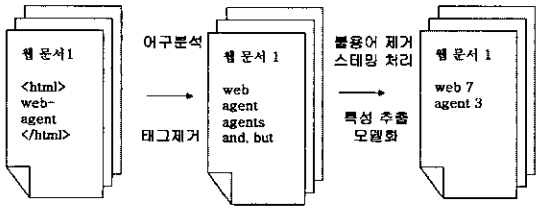
또한 분류가능 클래스들을 확장하고 대응되는 양질의 훈련 예를 확보하기 위해 인터넷 디렉토리 서비스를 제공하는 Yahoo 사이트로부터 교육(education), 과학(science), 컴퓨터와 인터넷(computer & internet) 등을 포함하는 14개의 최상위 클래스들에 대한 관련 웹 문서들을 추가적으로 수집한다. 그리고 이와 같이 분류된 북 마크들과 Yahoo 사이트를 이용해 훈련 문서들을 수집할 때 각 분류 클래스 별로 충분한 훈련 문서들을 확보하기 위해 분류 대상 문서와는 달리 북 마크가 가리키는 동일 웹 사이트 내의 이웃한 다른 웹 문서들도 함께 가져와서 이용한다. 이때 (그림 3)과 같이 사용자가 제공한 분류 클래스들과 Yahoo의 분류 클래스들 간에 분류 클래스별 훈련 문서들의 수를 균등하게 유지하도록 한다.



(그림 3) 웹 문서의 수집

3.5 웹 문서의 전처리

수집된 웹 문서들에 대해 문서 분류 학습법을 적용하기 위해서는 각 훈련 문서와 분류 대상 문서들을 적절한 특성 단어(feature word)들에 기초한 벡터 모델로 표현하여야 한다. 이를 위해서는 각 웹 문서를 표현하는데 중요한 역할을 하는 의미 있는 특성 단어들을 추출하는 것이 매우 중요한데, 이를 위해 먼저 각 웹 문서에 대한 전처리 과정(preprocessing)이 필요하다. 전처리 과정에서는 각 웹 문서를 구성 단어별로 나누는 작업과 더불어 웹 문서에서 태그(< >)를 제거하는 작업, and, but 등의 문서를 대표할 수 없는 단어들의 집합인 불용어를 제거하는 작업, 그리고 단어들의 어미 변화에 대한 처리인 스테밍(stemming) 처리작업 등이 이루어진다. (그림 4)는 이와 같은 웹 문서의 전처리와 이것에 기초한 문서 모델화 과정을 보여주고 있다.



(그림 4) 웹 문서의 전처리와 문서 모델화

3.6 특성 추출 및 모델화

특성 추출(feature extraction)과정은 분류 학습을 위해 각 문서들을 표현하는데 이용할 키워드(keyword)들을 결정하는 과정이며, 문서 모델화(document modeling) 과정은 정해진 특성단어에 기초하여 각 문서를 특성단어의 출현 유무, 빈도수(frequency), 혹은 가중치(weight) 등으로 표현하는 과정이다[15]. 특성추출과 문서 모델화 방법은 적용할 분류 학습기법과 더불어 문서 분류 성능에 가장 큰 영향을 주는 중요한 결정이 된다. 특히 특성 추출과 문서 모델화는 문서 분류 외에 정보검색(information retrieval), 정보여과 및 융합(information filtering and fusion) 등 다양한 분야에서 폭 넓게 이용되기 때문에 기존에 많은 선행 연구들이 있어 왔다.

문서들을 표현할 특성 단어들을 정하는 가장 기본적인 방법은 그 문서집합을 구성하는 모든 단어들의 집합(vocabulary)을 전부 특성 단어로 사용하는 것이다. 그러나 이러한 방법은 문서의 수에 비해 특성 단어의 수가 너무 많아져 - 경우에 따라서는 문서개수가 2,000~2,500일 때 서로 다른 구성 단어의 개수는 문서 개수의 10배인 20,000여 개를 상회한다 - 전체적으로 필요한 계산양이 많아질 뿐 아니라 분류에 영향을 주지 못하는 많은 수의 특성 단어로 인해 오히려 분류 성능이 낮아지기도 한다. 따라서 일반적으로 문서들을 구성하는 모든 단어들의 집합으로부터 분류에 큰 영향을 줄 수 있는 의미 있는 단어들만을 일부 특성 단어들로 선택하여 이용한다. 이와 같은 의미에서 특성 추출(feature extraction)을 특성 선택(feature selection), 차원 감소(dimension reduction) 등으로도 불린다. 특성 단어 선택을 위한 다양한 방법들이 그 동안 제안되었으나, 본 에이전트 시스템에서는 정보 이론(Information Theory)에 입각해 엔트로피(entropy) 변화량이 큰 단어들을 특성 단어로 선택하는 정보 획득(Information Gain)방법을 사용한다[16].

$$V = \{w_1, w_2, \cdots, w_n\} \tag{7}$$

$$InforGain(w_k) = P(w_k)\sum_i P(c_i \mid w_k)\log\frac{P(c_i \mid w_k)}{P(c_i)} + P(\overline{w_k})\sum_i P(c_i \mid \overline{w_k})\log\frac{P(c_i \mid \overline{w_k})}{P(c_i)} \tag{8}$$

문서들을 구성하는 전체 단어집합(V)이 식 (7)과 같이 총 n개의 단어들로 이루어져 있을 때, 각 단어 w_k에 대해 식 (8)과 같은 방식으로 정보 획득량을 계산하여 그 중 정보 획득량이 큰 L개의 단어만을 선택하여 식 (9)와 같은 특성 단어들의 집합을 구성한다.

$$K = \{w_1, w_2, w_3, \cdots, w_L\}, \quad K \subset V \tag{9}$$

선택된 특성 단어들의 집합으로 각 문서에 대한 모델을 만들 때 가장 많이 이용되는 문서 모델에는, 특성 단어의 출현

유무만으로 식 (10)과 같이 각 문서를 표현되는 이진 속성 벡터(vector of binary attributes) 방식, 특성 단어의 빈도수(frequency)로 식 (11)과 같이 표현하는 *Bag of Words* 방식, 특성단어별 TFIDF 가중치로 표현하는 가중치 벡터(weight vector)방식 등이 있다.

$$d_i = (1, 0, 1, \cdots, 1) \tag{10}$$

$$d_i = (2, 0, 1, \cdots, 3) \tag{11}$$

본 시스템에서는 TFIDF 분류 학습법을 위해서는 각 문서들을 TFIDF 가중치 벡터모델로, 나이브 베이지안과 K-NN 분류 학습법을 위해서는 Bag of Words 모델로 표현하였다.

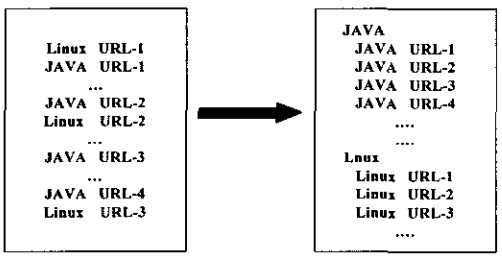
3.7 학습 및 분류

본 에이전트 시스템에서는 문서 분류를 위해 확률을 이용한 대표적인 교사학습(supervised learning) 알고리즘인 나이브 베이지안(naive Bayesian) 학습기법을 이용한다. 그러나 이외에도 K-NN 기법과 TFIDF 기법도 선택적으로 사용될 수 있도록 구현하였다. 원래 나이브 베이지안 분류학습법은 식 (3)에 의해 조건부 확률이 가장 큰 클래스로 문서를 분류한다. 하지만 본 시스템에서는 다른 클래스들에 비해 상대적으로 비록 조건부 확률이 가장 큰 경우라 하더라도 그 차이가 크지 않거나 조건부 확률의 절대치가 너무 낮은 경우 - 예컨대 조건부 확률의 최대치가 0.2~0.3 이하인 경우 - 에는 분류 결과에 대한 정확도와 신뢰도가 낮기 때문에 무리하게 주어진 클래스 중 하나에 자동으로 배정하지 않고 식 (12)와 같이 별도의 클래스 Others에 배정하여, 후에 사용자가 직접 분류할 수 있도록 분류 결정을 양도하게 된다. 식 (12)에서 임계치(threshold) T는 사전에 사용자가 정해줄 수 있다.

$$c(d_i) = \begin{cases} \arg\max_{c_j \in C} P(c_j \mid d_i) & \text{if } \max_{c_j \in C} P(c_j \mid d_i) \geq T \\ c_{Others} & \text{otherwise} \end{cases} \tag{12}$$

3.8 정렬 및 새로운 북 마크 파일 생성

북 마크를 대신하는 각 웹 문서에 대한 모든 분류 작업이 완료되면, 먼저 분류 결과에 따라 동일한 클래스에 배정된 북 마크들끼리 모으는 (그림 5)와 같은 정렬작업을 수행한다.

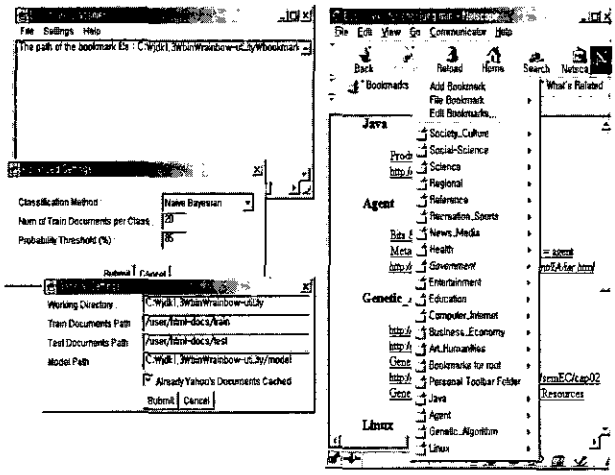


(그림 5) 정렬 작업

이어서 이렇게 새롭게 분류된 결과물은 이미 사용자에 의해 분류된 북 마크들과의 통합작업을 거쳐 북 마크 폴더별로 북 마크들이 분류된 새로운 북 마크 파일을 생성한다. 새로운 북 마크 파일 생성을 위해서는 3.2절에서 살펴본 북 마크 파일 형식을 따른다.

4. 시스템의 구현

북 마크 자동 분류 에이전트인 BClassifier는 300Mhz 펜티엄II 프로세서, 128M 주기억 장치와 리눅스(Linux) 환경의 컴퓨터에서 자바(java) 프로그래밍 언어를 사용하여 구현하였다. (그림 6)은 BClassifier의 실행 화면의 일부를 보여준다.



(그림 6) BClassifier의 실행 화면

(그림 6)의 좌측 상단에 위치한 사용자 인터페이스 주윈도우는 사용자 선택 메뉴들과 시스템의 현재 실행 상태를 보여준다. 그 아래에 위치한 두 개의 윈도우는 북 마크 파일의 위치와 작업 폴더 위치, 훈련 및 분류대상 문서들의 경로를 설정하는 일반설정(general settings) 윈도우와 분류 학습 기법 선택, 클래스 당 훈련 문서의 개수, 분류 임계치 등을 설정하는 고급설정(advanced settings) 윈도우이다. (그림 6)의 우측에는 BClassifier 에이전트가 북 마크 파일에 대한 분류 및 정렬 작업을 마친 후 그 결과를 네스케이프 웹 브라우저에서 보여주고 있다.

BClassifier 에이전트의 실행 속도는 실제 분류작업 자체보다는 훈련 문서 및 분류대상 문서들을 인터넷에서 수집하고 이들을 모델화하는데 필요한 지연시간에 많이 의존하였다. 따라서 Yahoo 사이트로부터 가져오는 14개 분류 클래스와 훈련 문서들은 분류 작업이 일어날 때마다 매번 새로 수집하지 않고 최초의 분류 작업 때에 일괄적으로 한번 수집한 것을 계속 사용함으로써 실행 속도를 높였다. 또한 반드시 사용자에게 확인 후에 기존의 북 마크 파일을 새로 생성

된 북 마크 파일로 대체하도록 하고, 대체한 후에도 기존의 북 마크 파일은 백업용으로 보존함으로써 추후 복구를 용이하게 하였다. 또 BClassifier는 필요에 따라 사용자가 정해준 일정한 시간 간격(time interval)마다 주기적으로 자동 분류 작업을 수행하도록 설정할 수 있다.

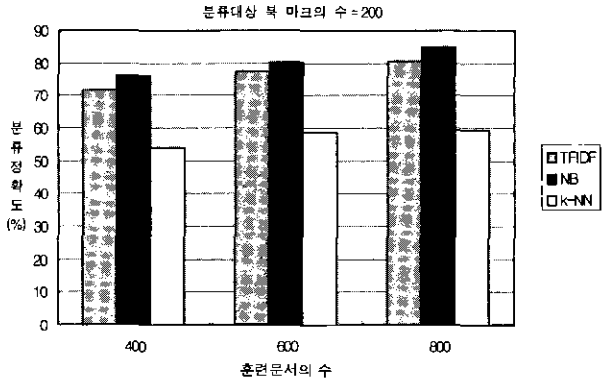
5. 실험 및 평가

5.1 실험 목적 및 방법

본 연구에서는 북 마크 자동 분류 에이전트인 BClassifier의 분류 성능을 분석하기 위한 몇 가지 실험을 전개하였다. 그리고 이 실험에서는 BClassifier에 구현된 3가지 서로 다른 분류 학습 기법들간의 분류 정확도도 함께 비교하는 것을 실험 목적으로 삼았다. 실험 환경으로는 300MHz 펜티엄 II 프로세서와 256MB의 주 기억공간을 사용하는 리눅스 환경과 넷스케이프 네비게이터 웹브라우저가 실험에 이용되었다. 실험에는 이미 분류된 북 마크들과 분류되지 않은 북 마크들을 담고 있는 임의의 북 마크 파일을 사용하였으며, 훈련 문서들은 Yahoo 사이트의 14개 최상위 클래스와 사용자가 정의한 6개의 클래스들로부터 각 클래스 당 40개씩, 총 800개의 웹 문서를 실험에 사용하였다. 실험 방법은 분류되지 않은 북 마크를 클래스 당 10개씩 임의로 선정하여 총 200개의 북 마크를 분류대상으로 삼았으며, 클래스별 훈련 문서를 20개, 30개, 40개로 점차 늘여가면서 이들에 대해 서로 다른 3가지 분류 학습법인 나이브 베이지안, TFIDF, K-NN 등의 3가지 분류 학습법을 차례로 적용시켜 분류 정확도(classification accuracy)를 측정하였다.

5.2 실험 결과

(그림 7)는 본 연구의 실험 결과를 보여주는 차트(chart)이다. BClassifier는 사용하는 분류 학습 기법에 따라 차이가 있기는 하지만 전체적으로 70%~80% 정도의 높은 분류 정확도를 보여주었다. 특히 예상한대로 훈련 문서의 수가 증가할수록 분류 성능도 조금씩 따라 증가하는 것을 발견할 수 있었다. 분류 학습 기법들간의 분류 성능의 차이도 비교적 뚜렷이 나타났는데 본 에이전트시스템에서 기본 분류방식으로 채택하고 있는 나이브 베이지안 학습법이 80%대의 가장 높은 분류 성능을 보여주었다. 이에 반해 k-NN 학습법은 기대와는 달리 50%~60%의 가장 낮은 분류 성능을 나타냈으며, TFIDF 학습법은 비교적 나이브 베이지안 학습법에 필적하는 좋은 성능을 보여주었다. 한편, 분류 학습 기법들간의 분류 속도 면에서는 개체기반 학습법(instance-based learning)의 하나로서 분류 당시에 많은 계산시간(computation time)을 필요로 하는 k-NN 학습법이 나머지 두 학습법에 비해 매우 느리게 나타났다. 그러나 전체적으로 BClassifier의 수행시간은 분류 자체에 필요한 지연시간보다는 훈련문서 및 분류대상 문서 수집에 필요한 인터넷상의 지연시간이 가장 큰 영향을 미쳤다.



(그림 7) 분류 성능 실험 결과

6. 결 론

본 논문에서는 기계 학습법을 적용하여 자동으로 북 마크들을 카테고리별로 분류하여 정렬해주는 개인화된 에이전트 시스템인 BClassifier를 설계하고 구현하였다. 이 시스템의 특징은 사용자가 이미 주제영역에 따라 북 마크 폴더별로 분류해놓은 북 마크들을 훈련 예로 사용하며, 또한 북 마크 자체를 분류하기보다는 북 마크가 가리키는 웹 문서들을 대신 사용하여 분류 작업을 한 다음 그 결과를 이용하는 방식을 취하였다. 또 사용자가 제공해줄 수 있는 분류 클래스와 훈련 예의 한계를 극복하기 위해 Yahoo 사이트의 최상위 14개 클래스들과 해당 웹 문서들을 추가적으로 이용한다. 본 논문에서는 구현된 BClassifier 에이전트의 분류 성능을 평가하는 실험을 통해 에이전트의 전체적으로 높은 분류 성능을 입증하였고 특히 분류 학습기법 중 나이브 베이지안 학습 기법의 우수성을 확인하였다. BClassifier의 성능과 효용성을 높이기 위해서 앞으로 시행되어야 할 향후 연구과제로, 분류 클래스들간의 계층관계 및 중복관계에 대한 해결, 사용자로부터 자동 분류 결과에 대한 직접적인 피드백(feedback)을 통한 학습 개선, 명확한 훈련 예를 확보할 수 없는 경우를 대비하기 위한 비교사 학습기법(unsupervised learning)에 대한 도입 등을 검토하고 있다.

참 고 문 헌

[1] Tom Mitchell, 'Machine Learning', McGraw Hill international Edition, 1995.

[2] D. D. Lewis and M. Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization," Proceeding of the 3rd Annual Symposium on Document Analysis and Information Retrieval, pp.81-93, 1994.

[3] Y. Yang and X. Liu. "A Re-Examination of Text Categorization Methods," Proceedings of SIGIR-99, 1999.
[4] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. "A Bayesian Approach to Filtering Junk E-Mail," Proceedings of AAAI-98 Workshop on Learning for Text Categorization, 1998.
[5] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," Proceedings of AAAI-98 Workshop on Learning for Text Categorization, 1998.
[6] T. Joachims,"A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," Proceedings of the 14th International Conference on Machine Learning ICML97, pp.143-151, 1997.
[7] Jeffrey M. Bradshaw, 'Software Agent', AAAI Press / The MIT Press, 1997.
[8] Stuart Russell and Peter Norvig, 'Artificial Intelligence : A Modern Approach', Prentice Hall, 1995.
[9] D. Mladenic, "Personal WebWatcher : Design and Implementation," Technical Report IJS-DP-7472, School of Computer Science, Carnegie-Mellon University, Pittsburgh, USA, October, 1996.
[10] B. Krulwich and C. Burkey, "The InfoFinder agent : Learning user interests through heuristic phrase extraction," IEEE Experts, Vol.2, No.5 pp.22-27, 1997.
[11] M. Pazzani and D. Billsus, "Learning and Revising User Profiles : The Identification of Interesting Web Sites," Journal of Machine Learning, Vol.27, No.3, pp.313-331, 1997.
[12] L. Chen and K. Sycara, "WebMate : A Personal Agent for Browsing and Searching," Proceedings of the 2nd International Conference on Autonomous Agents and Multi-Agent Systems, pp.132-139, 1998.
[13] P. Maes, "Agents That Reduce Work and Information Overload," Communications of the ACM, Vol.37, No.7, pp.30-40, 1994.
[14] B. Sheth and P. Maes, "Evolving Agents for Personalized Information Filtering," Proceedings of the 9th IEEE Conference on AI for Applications, 1993.
[15] D. D. Lewis, "Feature Selection and Feature Extraction for Text Categorization," Proceedings of Speech and Natural Language Workshop, pp.212-217, 1992.
[16] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," Proceedings of the 14th International Conference on Machine Learning, pp.412-420, 1997.

김 인 철

e-mail : kic@kyonggi.ac.kr
1985년 서울대학교 수학과 졸업(학사)
1987년 서울대학교 대학원 계산통계학과 (이학석사)
1995년 서울대학교 대학원 전산과학과 (이학박사)
1989년~1995년 경남대학교 전산통계학과 조교수
1996년~현재 경기대학교 정보과학부 전자계산학전공 부교수
관심분야 : 지능형 에이전트, 분산인공지능, 데이터마이닝

조 수 선

e-mail : scho@etri.re.kr
1987년 서울대학교 계산통계학과 졸업 (학사)
1989년 서울대학교 대학원 계산통계학과 졸업(이학석사)
1989년~1994 (주)웅진미디어 CBE개발부 연구원
1994년~현재 한국전자통신연구원 컴퓨터 소프트웨어연구소 선임연구원
관심분야 : 임베디드 시스템, 실시간OS, 인터넷 소프트웨어