온톨로지 기반의 웹 페이지 분류 시스템

최 재 혁[†] · 서 혜 성[††] · 노 상 욱[†††] · 최 경 희[††††] · 정 기 현[†††††]

요 약

본 논문은 온톨로지(ontology)에 기반한 자동화된 웹 페이지 분류 시스템을 제안한다. 웹 페이지의 분류를 위하여 첫 번째 단계에서는 각 웹 페이지가 속한 범주(category)를 대표할 수 있는 단어를 선정하며, 이를 위하여 단어빈도와 문서빈도를 곱한 값을 계산한다. 두 번째 단계에서는 첫 번째 단계에 의해 선택된 단어의 정보이득(information gain)을 계산해 분류 확률이 높은 단어를 우선적으로 선정한다. 두 단계를 통하여 선정된 단어들과 웹 페이지의 분류 정보를 가지고, 기계학습에 의하여 컴파일된 규칙(compiled rules)을 생성한다. 생성된 규칙은 임의의 웹 페이지들을 도메인 온톨로지에 의해 정의된 범주 별로 분류할 수 있도록 한다. 본 논문의 실험에서는 주어진 웹 페이지 집합에서 각 범주 별로 평균 240개의 단어로부터 78개의 단어를 결과적으로 선정하였으며, 이를 바탕으로 웹 페이지 분류 규칙을 생성하였다. 실험 결과에서 제안한 시스템의 평균 분류 정확도는 약 83.52%로 측정되었다.

Web Page Classification System based upon Ontology

Jaehyuk Choi[†] · Haesung Seo[††] · Sanguk Noh[†††]
Kyunghee Choi[††††] · Gihyun Jung[†††††]

ABSTRACT

In this paper, we present an automated Web page classification system based upon ontology. As a first step, to identify the representative terms given a set of classes, we compute the product of term frequency and document frequency. Secondly, the information gain of each term prioritizes it based on the possibility of classification. We compile a pair of the terms selected and a web page classification into rules using machine learning algorithms. The compiled rules classify any Web page into categories defined on a domain ontology. In the experiments, 78 terms out of 240 terms were identified as representative features given a set of Web pages. The resulting accuracy of the classification was, on the average, 83.52%.

키워드 : 웹 페이지 분류(Web Page Classification), 온톨로지(Ontology), 정보이득(Information Gain), 기계학습(Machine Learning)

1. 서 론

현대 사회는 인터넷의 발달로 새로운 방식의 사무형태를 자리잡아 가고 있다. 점차 다양하고 신속한 정보를 필요로 하는 시대적 요구에 부응하여 나타난 인터넷이라는 매체는 저렴한 비용으로 다양한 정보를 제공하며, 여타의 매체보다 신속하게 정보를 전달할 수 있다. 따라서 기업들은 인터넷의 사용이 필수 불가결하게 되었으며 이에 따른 투자를 점차 늘리고 있다. 그러나 그 정보의 다양함 가운데는 업무를 방해하는 유해한 요인들도 있어 이 때문에 오히려 업무의 효율성이 저하되는 사례들이 발견되고 있다[3]. 이런 요인들은 자원 운용 측면에서 볼 때, 불필요한 컴퓨팅 자원의 낭비와 통신 대역의 손실을 유발한다. 때문에 많은 기업들은 업무에 적합하지 않은 웹 페이지들을 탐지하고 차단하기 위한 방법들에 대한 관심을 가지게 되었다. 본 논문에서는 일반적인 네트워크 환경을 통해 사내망과 같은 지역 네트워크 안으로 전달되는 스트림들로부터 부적합한 웹 페이지들을 판별하여 온톨로지(ontology)[10, 11]에 기반한 클래스로 분류하는 방법을 소개한다.

본 논문이 제안하는 웹 페이지 분류 시스템은 크게 세 개의 모듈로 구성되어 있다. 그 세 개의 모듈은 패킷 수집 모듈, 프로토콜 분석 모듈, 콘텐트 분류 모듈이다. 패킷 수집 모듈은 TCP 기반의 네트워크에서 서버와 클라이언트 간의 세션으로부터 패킷을 수집한다. 프로토콜 분석 모듈은 수집된 패킷의 헤더 정보를 분석하고 패킷이 사용되는 응용 프로그램에 적합한 형태로 재구성한다. 예를 들어, 웹 서비스의 경우 웹 서버가 지원하는 압축 방식으로 웹 페이지를 제공하기도 하는데, 이와 같은 웹 페이지를 분석하기 위해서 프로토콜 분석 모듈이 웹 서버가 지원하는 압축 방식을 식별하여 해제

※ 이 논문은 2003년도 한국학술진흥재단의 지원과 국가지정연구실 사업의 지원에 의해 연구되었음(KRF-2003-041-D20465, KISTEP National Research Laboratory program).
† 정 회 원 : 삼성전자 네트워크 사업부
†† 정 회 원 : 아주대학교 정보통신전문대학원 정보통신공학과
††† 정 회 원 : 가톨릭대학교 컴퓨터정보공학부 교수
†††† 정 회 원 : 아주대학교 정보통신전문대학원 교수
††††† 정 회 원 : 아주대학교 전자공학부 교수
논문접수 : 2004년 4월 29일, 심사완료 : 2004년 8월 12일

하는 역할을 한다. 콘텐트 분류 모듈은 프로토콜 분석 모듈이 재구성한 웹 페이지의 내용을 분석하여 적합한 범주로 분류한다.

웹 페이지의 내용을 식별하기 위해서 본 논문에서는 단어빈도(Term Frequency : TF)와 문서빈도(Document Frequency : DF)[1, 18]를 곱한 값과 정보이득(information gain) 계산 방법[20]을 결합한 방법을 제안한다. 그리고 제안된 방법론에 의해 선택된 단어를 기반으로 하여 임의의 웹 페이지를 도메인 온톨로지 내에 있는 범주로 분류한다. 이 과정에서 실질적인 분류는 기계학습(machine learning)에 의해 생성된 분류 규칙을 사용하여 웹 페이지와 미리 정의된 범주 간의 관련성을 식별함으로써 가능하게 되는 것이다.

웹 페이지의 분류를 위하여는 웹 페이지가 속하는 범주를 우선적으로 정의해야 하며, 이를 위하여 본 논문에서는 온톨로지(ontology)를 구성한다. 온톨로지는 객체와 특성들을 계층적으로 구조화시켜 모델링하는 특징이 있으며, 개념의 확장이나 진행을 지속적으로 유지시켜 줄 수 있다. 그러나, 변화되는 온톨로지 구성 요소를 자주 갱신하지 않으면 웹 페이지 분류에 대한 정확도가 낮아지는 문제가 발생한다. 따라서 제안하는 시스템은 보다 정확한 온톨로지를 유지하기 위하여, 웹 페이지들의 범주를 계층적으로 구성하고 각 범주를 대표할 수 있는 단어들을 계속적으로 갱신한다.

제안하는 웹 페이지 분류 시스템은 실시간으로 웹 페이지를 분류하여 검사 중인 웹 페이지가 지역 네트워크에 부적합한 범주에 속할 경우 별도의 보고서를 생성할 수 있을 것으로 기대한다. 또한 웹 페이지 분류 시스템은 향후에 네트워크의 사용이 안전하게 되고 있음을 보증하는데 필수 불가결한 요소가 될 것임을 확신한다.

본 논문의 다음 장에서는 온톨로지 등을 이용한 웹 페이지 분류 방법과 관련된 연구들에 대해 논하고, 3장에서는 온톨로지를 구성하는 단어 선별 과정에 대한 자세한 설명을 한다. 4장에서는 웹 페이지 분류 시스템에 대한 시스템 구조를 설명하고, 5장에서는 본 논문에서 수행한 시뮬레이션 결과에 대해 논한다. 끝으로 결론에서 실험 결과를 정리하고 향후 연구 과제에 대해 언급한다.

2. 관련 연구

지금까지 온톨로지를 기반으로 하여 웹 페이지를 분류하려는 많은 연구가 있어왔다[6, 8, 12, 13]. Prabowo[13]와 그 외의 연구에서 온톨로지는 "특정 도메인의 개념적 인스턴스이며, 다른 인스턴스와 구별될 수 있는 개체(entity)"라고 정의했다. Prabowo는 듀이 십진 분류법(DDC)과 미 의회 도서관 분류법(LCC)을 고려하여 웹 페이지를 분류했는데 이 과정에서 온톨로지를 구축하였다. 이 방법은 기존의 접근법에 비해 몇 가지 특징이 있다. 그 특징은 분류를 위해 백과사전(thesaurus)이나 사전을 이용하지 않고, 온톨로지를 사용했다는 점과 웹 페이지의 텍스트로부터 온톨로지를 구축했다는 점, 그리고 온톨로지와 분류체계의 연관 관계를 구성한 점 등을 들 수 있다. 이 접근법의 단점은 비록 표준 분류 방법을 따르고 있지만, 분류에 대한 사용자의 복잡한 요구가 적절히 반영되기 어렵다는 것이다. 반면에, 본 논문에서 제안하는 방법은 온톨로지의 구성요소를 지속적으로 갱신하기 때문에 사용자의 요구에 보다 유연하게 대처하는 장점을 갖는다.

분류 규칙을 만들기 위해서는 기계학습에 적당한 크기의 특성들을 입력값으로 제공해야 한다. 제공할 특성을 선별하기 위한 방법으로 Ruger와 Gauch[17]의 연구에서는 단어빈도의 최소, 최대값 만의 임계값을 설정해 놓고, 그 사이의 단어들을 선택했다. 그들의 연구는 문서빈도에 의존적이어서 특성으로 선택된 단어들이 웹 페이지들의 특정 범주에서 자주 발견되기는 하지만, 그것을 다른 범주와 그것이 속한 문서의 범주를 구분하는 요소로 사용할 수는 없다. 그러나 본 논문은 문서빈도뿐만 아니라 단어빈도를 사용하여 각 범주를 대표하면서도 중요한 의미를 갖는 단어들을 선별할 수 있었다. 주목할 사항은 기존의 정보 검색에서 단어의 가중치를 계산하기 위해서 단어빈도(TF)와 역문서 빈도(IDF)를 이용하여 계산하나 본 시스템은 이를 변형한 TF/DF로써 각 단어의 가중치를 계산한다. TF/IDF가 각 문서마다 고유한 단어들에 가중치를 부여하는 반면 TF/DF는 각 문서 집합에 고유한 단어들에 가중치를 부여한다.

웹 페이지를 분류하기 위해서 대부분의 접근 방식들은 기계학습에 의한 접근 방식을 취하고 있다[16, 19, 22, 23]. Sebastiani[19]는 텍스트 기반의 분류법에서의 기계학습의 역할을 정리했다. Sebastiani에 의하면 기계학습 알고리즘을 통해 높은 성능을 얻기 위해서는 입력값으로서 선택되는 특성들의 크기를 줄이는 것이 관건이다. 본 논문의 관점에서는 Sebastiani의 연구가 도메인 온톨로지를 구축하지 않기 때문에 각 범주를 대표하기 위한 특성들이 중요하면서 가치 있는 것으로 선정되었는지 알 수 없다. 한편으로 Riboni[16]의 연구에서는 특성추출을 위해 정보이득, 단어 수(word frequency)[16] 그리고 문서빈도를 이용하였다. Riboni는 정보이득, 단어수, 문서빈도 각각에 의해 추출된 특성의 수를 변경하면서 기계학습 알고리즘을 적용하였다. 그 결과 정보이득에 의한 특성추출 방법이 분류 성능에 가장 좋은 영향을 미쳤다. 그러나 Riboni의 방법은 특성추출을 위해 사용한 각각의 방법을 개별적으로 사용하였기 때문에 결합된 방법이 분류 성능에 미칠 영향에 대해서는 연구가 미흡했다. 따라서 본 논문

에서는 특성들의 크기를 조절하여 사용할 수 있게 한 점에 대해서는 방법을 같이 하되, 분류의 정확도를 높이기 위하여 문서빈도, 단어빈도, 그리고 엔트로피 계산을 결합한 메커니즘을 제안한다.

3. 웹 페이지 분류를 위한 온톨로지

본 논문에서는 개념에 대한 계층적 구조를 갖는 온톨로지를 이용하여 웹 페이지를 모델링하고 분류하였다. 제안한 웹 페이지 분류 시스템의 온톨로지는 전체 개념의 계층구조에서 각각의 분류 범주를 정의하고 각 분류 범주를 대표하는 단어들을 포함한다. 따라서, 시스템의 분류 범주 각각이 주어진 웹 페이지를 어떤 기준(또는 단어)으로 분류하는지를 제시하며, 온톨로지의 분류 범주에 포함된 단어를 이용하여 기계 학습 규칙을 생성한다. 그러므로 주어진 도메인의 온톨로지를 정의하기 위하여 특정 범주(category)에 연관된 단어의 집합을 어떻게 선택하는가가 중요한 요인이 된다. 3장에서는 이러한 단어 집합의 선택 방법에 대하여 구체적으로 설명한다.

3.1 특성 추출

도메인 온톨로지를 구축하는 첫 번째 단계는 특정 범주를 대표할 수 있는 적절한 단어의 집합을 식별하는 일이다. 직관적으로 볼 때, 어떤 단어가 특정 범주에 속하는 한 웹 페이지 내에서 자주 발견되고 또한 그 특정 범주에 속하는 대부분의 웹 페이지들에서도 발견이 된다면 그 단어는 특정 범주를 대표한다고 할 수 있을 것이다. 예를 들어, "은행과 금융"이라는 범주를 포함하는 도메인 온톨로지가 있고, 'account'라는 단어가 앞서 설명한 대로 "은행과 금융" 범주 내의 특정 웹 페이지에 집중적으로 나타나고 또한 "은행과 금융" 범주 전체에서 골고루 발견된다면 'account'라는 단어는 "은행과 금융" 범주에 속하는 웹 페이지들을 대표하는 단어가 될 수 있다.

한 페이지에서 발견되는 단어의 빈도수와 특정 단어가 전체 문서에서 발견되는 빈도수는 정보검색 분야에서 사용되는 단어빈도(TF)와 문서빈도(DF)[1, 18]라는 용어로 대체하여 사용할 수 있다. 앞서 직관적인 개념에 의존했던 설명을 단어빈도와 문서빈도를 이용한 수식으로 설명하고자 한다면 다음과 같이 할 수 있을 것이다. 임의의 웹 페이지에 속하는 단어의 집합을 $I=\{1,2,\cdots,m\}$라 하고, 특정 범주에 속하는 웹 페이지들의 집합을 $J=\{1,2,\cdots,n\}$라 하자. 단어빈도와 문서빈도는 각 범주를 대표하는 단어에 대해 비례관계에 있으므로 이를 일반화하여 표현하면 아래 식과 같다.

$$W_{i,j}=\frac{TF_{i,j}}{\max_{k\in I} TF_{k,j}}\times\frac{DF_i}{n} \tag{1}$$

- $W_{i,j}$는 웹 페이지 $j\in J$에 속하는 단어 $i\in I$의 가중치이다.
- $TF_{i,j}$는 웹 페이지 $j\in J$에 속하는 단어 $i\in I$의 빈도이다.
- DF_i는 단어 i가 존재하는 웹 페이지의 개수이다.

위의 식 (1)은 특정 단어가 하나의 웹 페이지에서 다른 단어들 보다 많이 나타나고 또한 그 단어가 범주를 구성하는 대부분의 웹 페이지에서 나타난다면 그 가중치는 다른 단어들에 비해 크다는 것을 의미한다. 다른 단어들에 비해 더 큰 가중치를 가지는 단어는 그 단어가 속하는 범주에 대한 대표성이 더 크다는 것을 의미한다. 단어의 가중치를 계산함으로써 단어와 범주 간의 관련성을 측정해 볼 수 있다. 따라서 높은 가중치의 단어일수록 계층적 온톨로지 상에 특정 범주를 표현하는데 사용될 가능성이 높게 된다.

3.2. 엔트로피의 계산

우리는 특정 범주에 연관된 단어들을 가지고 임의의 웹 페이지를 온톨로지 기반으로 나눠진 범주들 중 하나의 범주로 분류해야 한다. 그러나 3.1절에서 계산된 단어들의 가중치는 그 단어가 해당 범주를 대표한다는 것을 나타내는 데는 유용하게 사용될 수 있지만, 범주를 분류하는데 있어 어느 정도 유용한지는 알 수 없다. 또한 유사 범주들간에는 같은 단어가 많이 나타나기 때문에 각 범주를 잘 설명하는 단어가 곧 해당 범주를 다른 범주와 잘 구분하는 단어가 될 가능성 또한 작아진다. 따라서 본 논문은 웹 페이지의 분류를 위해서 [20]에서 제안된 방법을 통해 단어의 엔트로피를 계산하는 방식을 도입한다. 단어의 엔트로피는 정확한 분류를 하는 데 필요한 정보의 기대 값을 제공한다. 낮은 엔트로피 값은 웹 페이지를 분류하는데 필요한 정보의 양이 적다는 것을 의미한다. 그러므로 보다 효율적인 분류를 위해서는 낮은 값의 엔트로피를 갖는 속성들을 사용해야 한다. 엔트로피의 계산은 다음과 같은 식으로 표현할 수 있다.

$$E^S=-\sum_{i=1}^{n} p_i\log_2(p_i) \tag{2}$$

- E^S는 확률 $p_1,\cdots,p_n$의 클래스들의 집합 S의 엔트로피이다.

예를 들어 A와 B라는 두 개의 클래스로 구성되는 집합이 있을 때, 집합 내의 모든 개체 수에 대한 A 클래스에 속하는 개체 수의 비율을 p_1이라 하자. 마찬가지로 B 클래스에 대한 p_2가 있다. 그러면 엔트로피는 $-(p_1\log_2 p_1+p_2\log_2 p_2)$에 의해 계산된다.

위에서 계산된 엔트로피를 가지고 속성의 분류 확률을 나타내는 정보이득 값을 계산할 수 있다. 정보이득에 대한 계산식은 아래와 같다.

$$Gain(\alpha) = E^{S} - \sum_{j=1}^{m} \left(\frac{|S_{a_j}|}{|S|} \times E^{S_{a_j}} \right) \quad (3)$$

- $Gain(\alpha)$는 속성 α에 대한 정보이득 값이다.
- S_{a_j}는 $j \in \{1, \cdots, m\}$의 값을 갖는 속성 α의 집합이다.

위 식에 사용된 속성이라는 것은 특정한 범주에 속하는 단어를 의미한다. 계산된 속성들의 정보이득 값을 통해 단어가 갖는 분류 확률의 순위를 매길 수 있으며, 정보이득 값에 근거해서 단어들을 선택할 수 있게 된다.

3.3. 단어 구성

웹 페이지를 분류하기 위해 속성들을 하나의 튜플로 조합시킬 필요가 있다. A를 속성들의 집합이라고 한다면, 속성의 선택은 아래와 같은 식을 적용하여 수행한다.

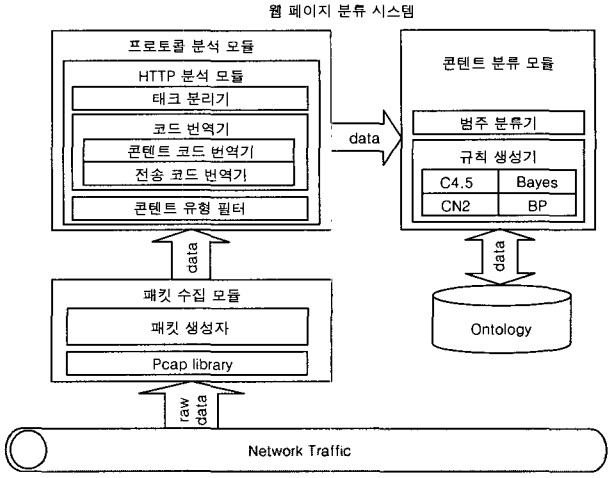
$$H_{\mu} = \{\alpha \mid \alpha \in A,\ Gain(\alpha) \geq \mu\} \quad (4)$$

임계값 μ는 각각의 범주에 있는 단어들 중 상대적으로 중요하지 않은 단어들을 걸러주는 역할을 한다. 위의 식을 수행한 결과로 만들어진 집합 $H\mu$는 μ가 높을수록 해당 범주와 관련이 깊은 속성들의 집합이 된다.

요컨대 자동화된 웹 페이지 분류를 위해서는 첫 번째로 TF/DF를 계산하여 해당 범주를 대표할 수 있는 단어들을 선정한다. 두 번째로 첫 번째 과정을 통해 선정된 단어들의 정보이득을 계산한 값을 이용하여 속성들의 우선순위를 정해 튜플을 구성한다. 세 번째로 구성된 튜플을 가지고 여러 기계학습 알고리즘[5, 9]에 대입해 분류 규칙의 집합을 만든다. 결과적으로 생성된 규칙을 통해 임의의 웹 페이지를 도메인 온톨로지가 정의하는 범주로 분류할 수 있도록 한다.

4. 분류 시스템의 구조

이번 장에서는 웹 페이지 분류 시스템에 대한 시스템 구조를 설명한다. (그림 1)은 웹 페이지 분류 시스템의 구조를 나타낸다. 웹 페이지 분류 시스템은 크게 세 개의 모듈 즉, 패킷 수집 모듈, 프로토콜 분석 모듈, 콘텐트 분류 모듈로 구성되어 있다. 시스템의 전반적인 흐름은 네트워크 트래픽 중 일부 트래픽을 패킷 수집 모듈로 수집한 후에 프로토콜 분석 모듈이 분석 가능한 형태로 바꾸고 그것을 콘텐트 분류 모듈이 온톨로지 데이터베이스를 기반으로 미리 정의된 범주 별로 분리하는 것이다.



(그림 1) 웹 페이지 분류 시스템의 구조

패킷 수집 모듈은 libpcap[7]을 이용하여 네트워크 트래픽 중에 TCP 80 포트를 통해 흐르는 스트림을 수집하여 패킷 생성자를 통해 구조체로 표현한 후 프로토콜 분석 모듈로 넘긴다.

〈표 1〉 HTTP 프로토콜의 Content-Encoding 유형

인코딩 방식	설 명
gzip	GNU zip 형식의 압축 포맷
compress	UNIX 파일 압축 프로그램의 압축포맷. LZW 라고도 함.
deflate	zlib과 deflate라는 압축 메커니즘이 결합된 형태
identity	기본 인코딩. 변환이 필요 없음.

프로토콜 분석 모듈은 현재 HTTP 프로토콜에 대한 분석만이 가능하므로 패킷 수집 모듈에서 수집된 데이터는 HTTP 분석 모듈로 전달된다. HTTP분석 모듈에서는 HTTP 프로토콜을 분석하여 최종적으로 분석 가능한 형태를 만든다. 최종적으로 분석 가능한 형태란 텍스트로만 이뤄진 데이터로 이를 얻기 위해서는 다음과 같은 과정을 거친다. 먼저 콘텐트 유형 필터에 의해 자료의 유형을 파악해야 한다. HTTP 프로토콜을 통해 전달되는 자료의 유형은 IANA[15]에 의해 등록되어있다. 이 유형 중 우리가 취해야 할 것은 웹의 콘텐트의 대부분을 차지하는 text/plain 유형과 text/html 유형이다. 이렇게 선별된 유형의 자료를 판독 가능한 형태로 바꾸는 작업을 해야 한다. 일부 HTTP 서버들은 <표 1>과 같은 코딩 방식 중 하나를 선택하여 인코딩하여 보내기도 한다.

그리고 <표 1>과는 성격이 다른 인코딩 방식이 사용되기

도 하는데 이는 문서의 안전한 전송을 위해 사용하는 인코딩 방식으로 일반적으로 "chunked"라고 불린다. 모든 인코딩이 해제되었다면 본 시스템에서는 <body> 태그 내의 내용만을 취급하므로 태그 분리기에 의해 태그로 사용되는 문자들을 모두 제거하고 본 내용만 추출한다. 위 과정을 통해 얻어진 콘텐트는 콘텐트 분류 모듈로 넘겨 해당 웹 페이지가 어떤 범주에 속하는지 분류한다.

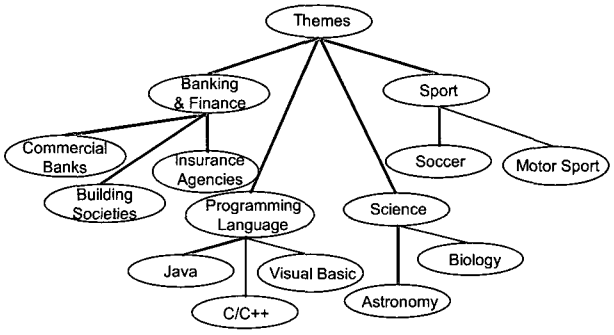
콘텐트 분류 모듈은 분류 규칙을 만드는 부분과 콘텐트의 범주를 분류하는 부분으로 구성되어 있다. 분류 규칙을 만드는 부분은 온톨로지 데이터베이스로부터 각 범주의 정보를 얻어와 3장에서 기술한 바와 같이 네 가지 기계학습 알고리즘을 통해 생성하도록 되어 있고, 콘텐트의 범주를 분류하는 부분은 생성된 규칙을 임의의 웹 페이지에 적용하여 온톨로지에 의해 정의된 범주 별로 분류하도록 되어 있다. 생성된 규칙 중 하나의 예로 CN2[2]가 생성한 규칙은 (그림 2)와 같다. (그림 2)에서 사용된 단어 'java', 'string', 'galaxi' 등은 3장에서 소개한 특성 추출 방법과 정보이득 계산에 의해 선출된 단어들로 이들을 기계학습의 입력값으로 사용한 것이다. CN2는 'IF … THEN … ELSE' 구조로 분류 규칙을 표현한다. 즉, 임의의 웹 페이지가 포함하는 단어들 중에 'java', 'string'이 없고 'galaxi'가 있다면 그것은 'SC' 즉, '과학(Science)' 범주에 속한다는 것을 뜻한다. 그 밖의 'SP', 'BF', 'PL'은 각각 '스포츠', '은행과 금융', '프로그래밍 언어'를 지칭한다.

```
CN2 Decision List :
IF      "java" = "false"
AND "string" = "false"
AND "galaxi" = "true"
THEN class = "SC"  [0 0 402 0]
ELSE
IF      "account" = "false"
AND "astronom" = "false"
AND "hole" = "false"
AND "footbal" = "true"
THEN class = "SP"  [0 0 0 583]
ELSE
IF      "bank" = "true"
AND "loan" = "true"
AND "distanc" = "false"
THEN class = "BF"  [405 0 0 0]
ELSE
IF      "financi" = "false"
AND "java" = "true"
AND "code" = "true"
AND "sky" = "false"
AND "manchest" = "false"
THEN class = "PL"  [0 552 0 0]
.....
```

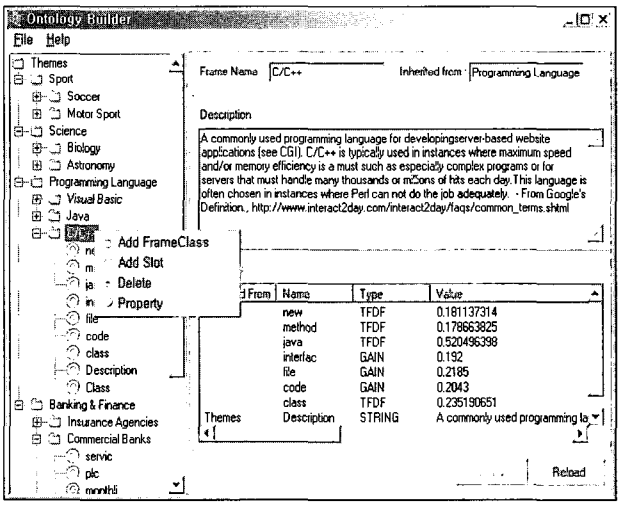
(그림 2) CN2에 의해 생성된 분류 규칙의 예

5. 실험 결과

우리는 자동화된 웹 페이지 분류 시스템의 분류 정확도에 대한 성능을 검증하기 위하여 Sinka[21]의 연구에서 제공하는 데이터 집합을 사용하였다. Sinka[21]가 사용한 데이터 집합은 웹 페이지 클러스터링 알고리즘을 평가하기 위해 제안한 평가 문서 집합으로 본 시스템의 웹 페이지 분류의 평가에도 적합하다고 판단된다. 실험을 위하여 1.6GHz의 펜티엄 IV PC를 사용하였다.



(그림 3) Sinka [21]가 제공한 네 가지 범주 : 은행과 금융, 프로그래밍 언어, 과학, 스포츠



(그림 4) 온톨로지 생성기를 이용하여 프로그래밍 언어 범주를 정의한 예제 화면

Sinka의 데이터 집합을 상위레벨에서의 범주로 나타내면 (그림 3)과 같다. 데이터 집합은 은행과 금융, 프로그래밍 언어, 과학, 스포츠로 총 네 개의 범주로 구성되어 있다. 주어진 데이터 집합에서 각 범주를 계층구조로 하는 온톨로지를 정의하기 위하여, (그림 4)와 같은 온톨로지 생성기(ontology builder)를 설계 및 구현하였다. 온톨로지 생성기를 이용하여 각 분류 범주에 속한 단어들을 정의하고 유지하

며, 온톨로지에 정의된 단어들을 이용하여 기계학습 규칙을 생성한다.

본 논문에서 제안한 단어 선택 방법의 유용성을 검증하기 위해 (1) 전체 범주의 분류를 위하여 각 범주 하위 클래스 중 하나씩만을 선택하여 분류성능을 테스트 하였으며, 또한 (2) 유사 범주의 분류를 위하여 은행과 금융 범주의 하위의 세 개의 클래스간의 분류 성능을 테스트 하였다. 마지막으로, (3) 종합적인 분류 성능의 테스트를 위해, 전체 범주와 유사 범주에 사용한 여섯 개의 클래스를 모두 포함하는 통합 범주의 분류 성능을 테스트 하였다. 전체 범주의 분류를 위하여 'Commercial Bank', 'Java', 'Astronomy', 'Soccer' 클래스를 선택하였으며, 유사 범주의 분류를 위하여 'Commercial Bank', 'Building Societies', 'Insurance agencies' 클래스를 선택하였다. 이들 클래스들은 (그림 3)에서 각각 굵은 선으로 표시되었다.

〈표 2〉 은행과 금융, 프로그래밍 언어, 과학, 스포츠 범주에 사용된 단어들의 가중치

전체 범주							
은행과 금융		프로그래밍 언어		과 학		스 포 츠	
단 어	가중치	단 어	가중치	단 어	가중치	단 어	가중치
account	0.2469	java	0.5205	galaxi	0.2141	cup	0.1946
bank	0.2432	class	0.2352	univers	0.2047	unit	0.1799
rate	0.1910	new	0.1811	star	0.1926	england	0.1786
servic	0.1721	method	0.1787	observ	0.1103	team	0.1618
interest	0.1471	code	0.1644	time	0.1078	fc	0.1552
insur	0.1391	com	0.1423	light	0.1061	footbal	0.1500
loan	0.1378	applet	0.1390	year	0.0992	club	0.1480
mortgag	0.1369	us	0.1281	scienc	0.0921	new	0.1361
card	0.1142	object	0.1202	imag	0.0916	citi	0.1331
save	0.1072	file	0.1159	space	0.0869	leagu	0.1247
inform	0.0997	page	0.0924	new	0.0855	town	0.1148
mai	0.0978	user	0.0914	telescop	0.0817	world	0.0919
year	0.0970	internet	0.0884	x	0.0803	game	0.0899
home	0.0896	program	0.0882	astronomi	0.0786	group	0.0859
person	0.0896	creat	0.0869	astronom	0.0782	y	0.0827

각 범주에 대한 대표적인 단어를 식별하기 위해서 각 범주에서 임의의 페이지 100개를 선별하여 3장의 식 (1)을 적용한다. 선택된 범주의 각 웹 페이지들은 전체 범주 분류의 경우 평균적으로 450개의 단어를, 유사 범주 분류의 경우 300개의 단어를, 통합 범주 분류의 경우 240개의 단어를 포함하고 있다. 이들에게 식 (1)을 적용한 결과는 <표 2>와 <표 3>에 있다. 결과를 살펴보면 전체 범주 분류의 경우, 은행과 금융 범주에서의 'account', 프로그래밍 언어에서의 'java', 과학의 'galaxi', 스포츠의 'cup'이 각각의 범주에서 가장 높은 가중치 값을 가졌다. 유사 범주 분류의 경우 상업 은행 범주에서 'account', 주택 금융 조합에서 'mortgag', 보험 대리점에서 'insur'가 각각의 범주에서 가장 높은 가중치 값을 가졌다. 또한 이 단어들은 직관적으로 볼 때 각각 해당 범주를 대표할 만한 단어들이라 할 수 있을 것이다. 또한 유사 범주 분류의 경우, 'bank', 'insur', 'mortag'와 같이 각 범주에 공통적으로 나타나는 단어들이 나타난다. 이것은 분류 범주 자체가 서로 의미적으로 중복되는 범위가 넓으므로 당연한 현상으로 생각된다. 통합 범주의 경우, 위 전체 범주와 유사 범주의 결과가 반복된 형태를 나타내어 생략하였다.

〈표 3〉 상업 은행, 주택 금융 조합, 보험 대리점에 사용된 단어들의 가중치

유사 범주					
상업 은행		주택 금융 조합		보험 대리점	
단 어	가중치	단 어	가중치	단 어	가중치
account	0.2469	mortgag	0.3817	insur	0.5080
bank	0.2432	rate	0.3448	polici	0.1594
rate	0.1910	societi	0.2875	servic	0.1543
servic	0.1721	interest	0.2496	cover	0.1452
interest	0.1471	account	0.2380	uk	0.1431
insur	0.1391	year	0.2143	provid	0.1291
loan	0.1378	build	0.1483	claim	0.1258
mortgag	0.1369	invest	0.1454	person	0.1025
card	0.1142	servic	0.1264	mai	0.0989
save	0.1072	fee	0.1198	protect	0.0987
inform	0.0997	loan	0.1185	inform	0.0965
mai	0.0978	term	0.1115	busi	0.0919
year	0.0970	charg	0.1106	compani	0.0873
home	0.0896	inform	0.1071	pleas	0.0870
person	0.0896	product	0.1011	benefit	0.0860

〈표 4〉 전체 범주 분류에 사용된 단어들에 대한 정보이득 값

전체 범주							
은행과 금융		프로그래밍 언어		과 학		스 포 츠	
단 어	정보이득	단 어	정보이득	단 어	정보이득	단 어	정보이득
bank	0.40	java	0.57	galaxi	0.38	footbal	0.39
insur	0.28	file	0.27	telescop	0.31	cup	0.38
plc	0.26	server	0.25	astronom	0.31	leagu	0.23
mortgag	0.26	class	0.24	astronomi	0.29	liverpool	0.22
loan	0.25	program	0.23	scienc	0.29	aston	0.22
payment	0.24	user	0.22	univers	0.27	tottenham	0.22
monthli	0.21	code	0.21	observ	0.26	chelsea	0.21
deposit	0.21	http	0.20	measur	0.25	arsen	0.21
account	0.20	window	0.20	light	0.24	villa	0.20
financi	0.20	html	0.20	space	0.23	rover	0.19
tax	0.20	text	0.15	sky	0.23	club	0.18
cash	0.20	creat	0.15	earth	0.22	ranger	0.18
branch	0.20	download	0.14	star	0.21	everton	0.18
invest	0.19	sourc	0.13	mass	0.18	leed	0.17
paid	0.19	england	0.13	scale	0.17	fc	0.17

〈표 5〉 유사 범주 분류에 사용된 단어들에 대한 정보이득 값

유사 범주					
상업 은행		주택 금융 조합		보험 대리점	
단 어	정보이득	단 어	정보이득	단 어	정보이득
bank	0.18	societi	0.30	bank	0.19
Plc	0.11	build	0.19	account	0.14
build	0.08	minimum	0.14	rate	0.12
card	0.08	mortgag	0.11	interest	0.11
euro	0.08	charit	0.10	loan	0.11
financ	0.07	princip	0.10	medic	0.10
overdraft	0.07	variabl	0.10	mortgag	0.09
access	0.06	gross	0.08	motor	0.08
faq	0.05	year	0.07	polici	0.07
internet	0.05	withdraw	0.08	claim	0.07
account	0.05	ombudsman	0.07	access	0.07
polici	0.05	valuat	0.07	underwrit	0.07
member	0.05	penalti	0.07	insur	0.07
save	0.04	rate	0.07	injuri	0.06
name	0.03	annual	0.07	pet	0.06

위의 과정에 의해 특정 범주와 연관되는 주요 단어들을 선정했다면, 선정된 주요 단어에 정보이득에 근거한 식 (3)을 적용하여 순위를 매긴다. 이것은 어떤 속성이 웹 페이지들을 분류하는데 좋은 단어가 되는 것인지 알아보기 위함이다. <표 4>와 <표 5>, <표 6>은 각 범주 별로 정보이득 값이 큰 단어들의 리스트를 보인 것이다. <표 5>를 살펴보면, 앞에서 언급하였듯이 유사 범주 분류의 각 범주에 속한 단어들이 서로 중복되는 것이 많은 반면, 정보이득으로 정제한 단어들은 중복된 단어의 개수가 현저히 줄어들었으며, 정제된 단어들이 (1)주어진 범주를 잘 설명하며 (2)다른 범주와 잘 구분하였다.

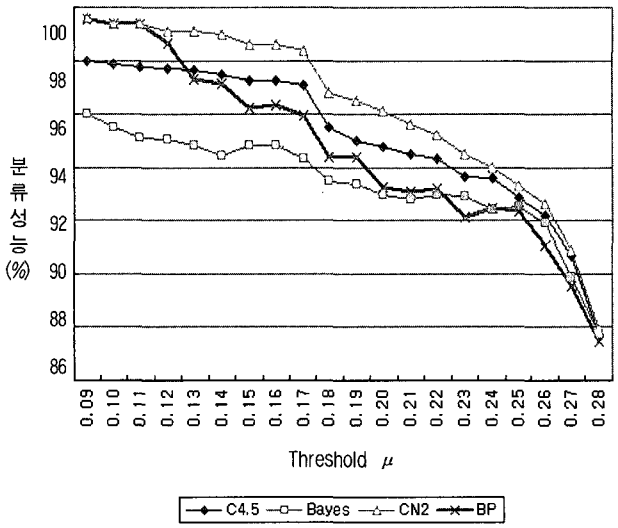
다음으로 기계학습에 사용할 단어를 선택하기 위해 순위가 매겨진 단어들에 대해 식 (4)를 적용해 임계값 μ 이상의 단어를 선택한다. 임계값 μ는 기계학습에 입력값이 될 단어의 개

〈표 6〉 통합 범주 분류에 사용된 단어들에 대한 정보이득 값

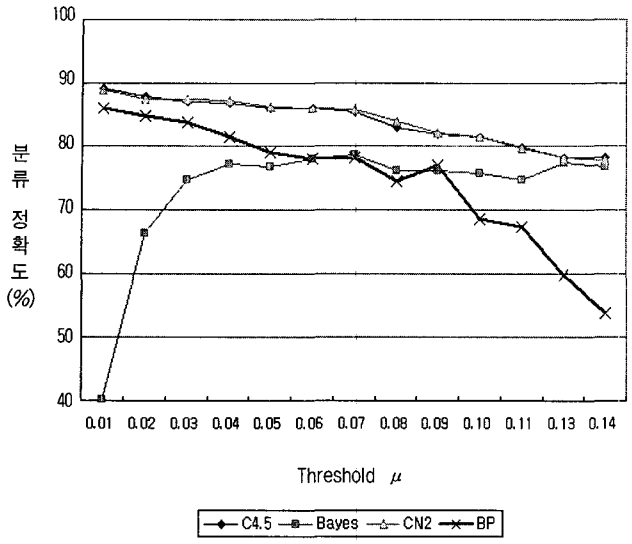
통합 범주											
상업 은행		주택 금융 조합		보험 대리점		프로그래밍 언어		과학		스포츠	
단 어	정보이득	단 어	정보이득	단 어	정보이득	단 어	정보이득	단 어	정보이득	단 어	정보이득
bank	0.19	societi	0.22	insur	0.12	java	0.43	galaxi	0.28	footbal	0.36
financ	0.10	mortgag	0.18	polici	0.08	tutori	0.24	univers	0.27	cup	0.35
plc	0.09	rate	0.15	cover	0.07	microsystem	0.10	observ	0.24	leagu	0.24
account	0.07	repay	0.14	premium	0.06	jdk	0.10	astronomi	0.24	team	0.17
loan	0.07	valuat	0.14	claim	0.06	applet	0.20	telescop	0.18	game	0.16
card	0.07	build	0.14	motorcycl	0.05	program	0.19	space	0.18	player	0.15
busi	0.07	invest	0.13	liabl	0.05	html	0.19	astronom	0.17	club	0.15
overdraft	0.06	borrow	0.12	uk	0.05	forum	0.19	densiti	0.15	england	0.15
credit	0.05	charg	0.11	medic	0.05	interfac	0.16	light	0.14	uefa	0.15
save	0.05	interest	0.10	expens	0.04	api	0.15	earth	0.13	match	0.14
telephon	0.04	regul	0.10	underwrit	0.04	user	0.15	cosmic	0.13	fa	0.14
invest	0.04	loan	0.10	motor	0.04	code	0.13	center	0.13	champion	0.13
deposit	0.04	capit	0.09	accid	0.04	creat	0.13	galact	0.12	belgium	0.13
branch	0.04	account	0.09	saveinsur	0.04	class	0.13	hubbl	0.13	goal	0.12
mortgag	0.04	payment	0.09	biba	0.04	window	0.12	sky	0.13	arsen	0.12

〈표 7〉 전체 범주 분류에 대한 임계값 μ의 변화에 따른 분류 성능 비교

μ	단어수	분류성능(단위: %)			
		C4.5	Bayes	CN2	BP
0.09	144	98.00	96.03	**99.61**	99.53
0.10	130	97.86	95.53	**99.39**	99.36
0.11	117	97.78	95.11	**99.39**	99.39
0.12	110	97.69	95.06	**99.11**	98.64
0.13	95	97.64	94.81	**99.11**	97.31
0.14	85	97.44	94.42	**99.00**	97.14
0.15	77	97.25	94.81	**98.61**	96.25
0.16	75	97.25	94.81	**98.61**	96.36
0.17	67	97.06	94.31	**98.39**	95.97
0.18	57	95.50	93.50	**96.81**	94.39
0.19	50	94.97	93.39	**96.50**	94.36
0.20	45	94.75	93.00	**96.11**	93.25
0.21	37	94.50	92.78	**95.61**	93.06
0.22	33	94.31	93.00	**95.19**	93.22
0.23	29	93.64	92.89	**94.50**	92.14
0.24	24	93.58	92.39	**94.00**	92.44
0.25	19	92.86	92.53	**93.31**	92.33
0.26	17	92.19	91.92	**92.61**	91.06
0.27	13	90.67	89.86	**90.89**	89.56
0.28	10	87.69	87.75	**87.89**	87.44



(그림 5) 전체 범주 분류에 대한 임계값 μ의 변화에 따른 분류 성능 비교



(그림 6) 유사 범주 분류에 대한 임계값 μ변화에 따른 분류 성능 비교

〈표 8〉 유사 범주 분류에 대한 임계값 μ의 변화에 따른 분류 성능 비교

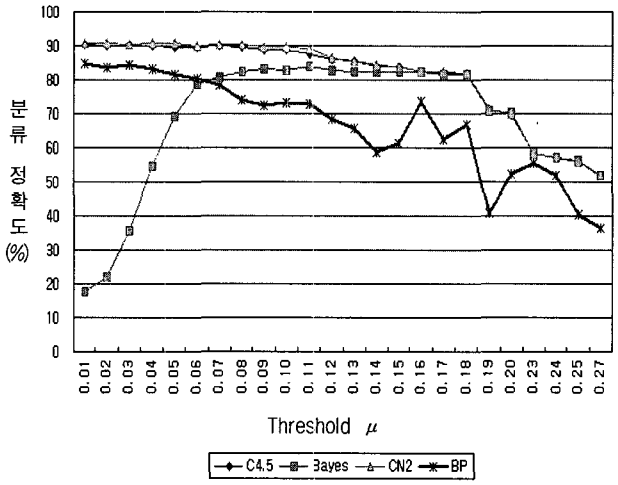
μ	단어수	분류성능 (단위: %)			
		C4.5	Bayes	CN2	BP
0.01	457	**89.21**	40.27	88.99	86.10
0.02	245	**87.84**	66.22	87.47	84.83
0.03	157	87.02	74.83	**87.39**	83.69
0.04	93	86.80	77.16	**87.36**	81.54
0.05	62	86.06	76.86	**86.25**	79.01
0.06	40	86.02	77.94	**86.06**	78.09
0.07	26	85.32	78.53	**85.73**	78.24
0.08	17	82.87	76.19	**83.91**	74.49
0.09	14	81.94	76.23	**82.13**	77.05
0.10	10	**81.57**	75.82	81.46	68.48
0.11	9	**79.79**	74.71	79.75	67.30
0.13	5	**78.16**	77.38	78.16	59.73
0.14	4	**78.16**	76.83	77.71	53.76

수를 조절하므로 분류성능에 큰 영향을 미친다. 예를 들어 임계값을 낮추면 선택되는 단어의 수가 증가하여 기계학습의 결과로 얻어지는 분류규칙의 분류성능이 높아진다. 그러나 분류규칙에 사용되는 단어의 수가 증가하는 만큼 분류를 위해 필요한 비용이 높아진다. 따라서 적은 비용을 들여 높은 분류 성능을 얻는 임계값을 확보하는 것이 중요하다. 우리는 전체 범주, 유사 범주, 통합 범주 별로 임계값 μ를 변동시키면서 분류성능을 측정한 결과를 각각 <표 7>과 <표 8>, <표 9>와 (그림 5)와 (그림 6), (그림 7)에 나타내었다. 표에서 강조된 수치는 같은 임계값 μ에서 가장 좋은 성능을 나타내었음을 의미한다.

(그림 5)는 μ값이 증가할수록 기계학습이 생성한 분류규칙의 분류 성능이 감소한다는 것을 보여준다. 이와 같은 경향은 (그림 6)과 (그림 7)에서도 나타난다. (그림 6)와 (그림 7)에서 μ값이 일정 수준 이상이 되면 분류 규칙이 학습 데이터에 과잉 맞춤(over fitting)되어 성능이 떨어지거나, 성능 향상이 크게 이루어지지 않는 것을 볼 수 있다. 또한 전체 범주 분류보다 유사 범주 분류의 경우 성능이 떨어지는 것을 볼 수 있다.

〈표 9〉 통합 범주 분류에 대한 임계값 μ의 변화에 따른 분류 성능 비교

μ	단어수	분류성능 (단위: %)			
		C4.5	Bayes	CN2	BP
0.01	1,172	90.52	17.60	**91.00**	84.65
0.02	905	90.18	21.91	**90.83**	83.61
0.03	681	90.24	35.71	**90.59**	84.27
0.04	502	90.09	54.95	**90.74**	83.02
0.05	392	89.65	69.25	**90.89**	81.59
0.06	286	89.59	78.90	**90.15**	80.51
0.07	229	89.98	80.77	**90.29**	78.38
0.08	173	89.44	82.55	**90.24**	73.81
0.09	132	88.76	83.15	**89.48**	72.29
0.10	104	88.72	82.74	**89.50**	73.11
0.11	78	87.76	84.02	**89.17**	72.92
0.12	59	85.91	82.65	**86.24**	68.31
0.13	42	85.53	82.29	**85.65**	65.64
0.14	30	83.83	82.52	**84.24**	58.71
0.15	29	83.85	82.42	**84.09**	61.12
0.16	22	**82.53**	82.26	82.44	73.48
0.17	18	82.13	81.53	**82.44**	62.33
0.18	16	**82.09**	81.66	82.07	66.64
0.19	12	**71.01**	70.85	70.79	40.88
0.20	10	69.72	**70.38**	69.90	52.29
0.23	9	58.27	58.40	**58.42**	55.60
0.24	7	57.14	**57.23**	57.12	51.99
0.25	5	56.23	**56.25**	56.12	40.51
0.27	4	51.90	**52.01**	51.90	36.38



(그림 7) 통합 범주 분류에 대한 임계값 μ의 변화에 따른 분류성능 비교

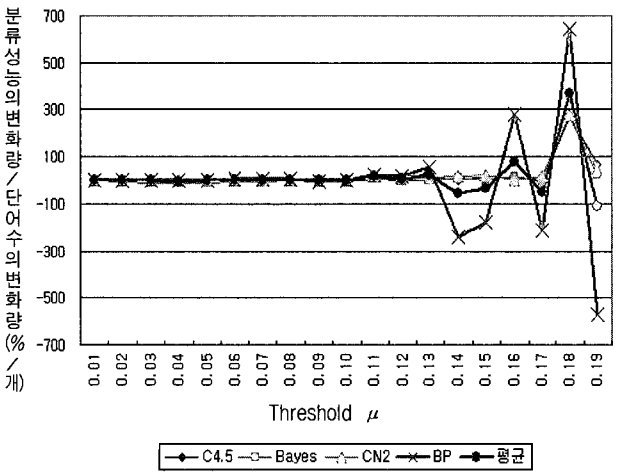
위 결과를 정리해 보면 무조건 많은 단어를 선택하는 것이 좋은 분류성능을 가져오지 않음을 알 수 있다. 이와 같은 현상을 <표 10>과 (그림 8)에 정리하였다. <표 10>은 μ값을 0.01부터 0.20까지 0.01씩 증가시킬 때 증가되는 분류 정확도를 추가로 선택되는 단어 개수로 나눈 값을 나타낸다. 굵은 글씨체로 강조된 수치는 기울기가 1.00이상인 것들을 표시한다.

선택된 단어의 수는 비용과 반비례 관계에 있기 때문에 단어의 수는 적으면 적을수록 좋다. 그러나 단어의 수를 줄이기 위해서는 임계값 μ를 높여야 하는데 이것은 분류 성능의 감소를 가져온다. 따라서 이를 절충하기 μ값에 따른 분류 성능의 변화 량이 가장 큰 구간 [0.11~0.12]에서 μ값 0.11을 선택하여 실험하는 것이 분류 성능과 비용 측면에서 유리하다고 판단된다. 따라서 우리는 기계학습 알고리즘의 입력값으로 쓰일 단어를 선택하기 위해 μ값 0.11을 선택하여 실험을 하였다.

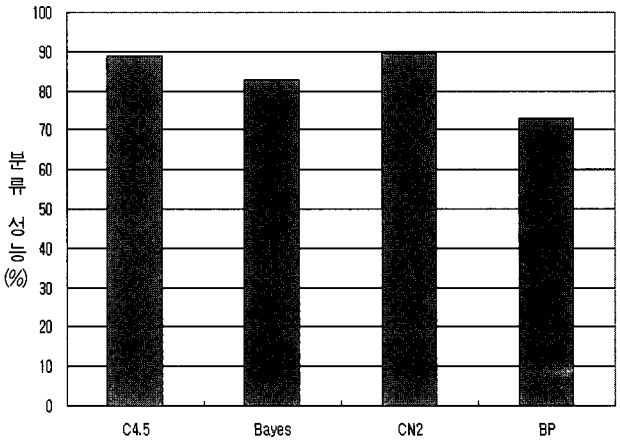
순위가 매겨진 단어들에 대해 식 (4)를 적용해 임계값 μ를 0.11로 정하고, 임계값보다 큰 정보이득 값을 갖는 단어들을 선택한다. 그 결과 범주 당 13개, 총 78개의 단어가 선택되었다. 마지막으로 선택된 단어들로 조합을 만들어 각각의 조합이 어떤 범주에 해당하는지 파악해 기계학습에 사용될 수 있게 만든다. 기계학습의 입력값으로 사용될 이 튜플의 크기는 엔트로피 계산에 의해 그 크기를 줄일 수 있는 것이다.

〈표 10〉 통합 범주 분류 상에서 μ값의 구간별 분류 성능의 변화량

μ 구간	단어수	단어 수의 변화량	분류 성능의 변화량(분류 성능의 변화량/단어 수의 변화량 %/개)				
			C4.5	Bayes	CN2	BP	평 균
0.01 ~ 0.02	1,172	267	0.12	-1.62	0.06	0.39	-0.26
0.02 ~ 0.03	905	224	-0.02	-6.16	0.11	-0.29	-1.59
0.03 ~ 0.04	681	179	0.08	-10.75	-0.08	0.70	-2.51
0.04 ~ 0.05	502	110	0.40	-13.00	-0.13	**1.30**	-2.86
0.05 ~ 0.06	392	106	0.05	-9.10	0.70	**1.02**	-1.83
0.06 ~ 0.07	286	57	-0.68	-3.28	-0.26	**3.74**	-0.12
0.07 ~ 0.08	229	56	0.96	-3.18	0.10	**8.16**	**1.51**
0.08 ~ 0.09	173	41	**1.67**	-1.45	**1.85**	3.71	**1.45**
0.09 ~ 0.10	132	28	0.13	**1.46**	-0.07	-2.93	-0.35
0.10 ~ 0.11	104	26	**3.70**	-4.91	**1.28**	0.73	0.20
0.11 ~ 0.12	78	19	**9.75**	**7.21**	**15.40**	**24.26**	**14.16**
0.12 ~ 0.13	59	17	**2.18**	**2.07**	**3.49**	**15.71**	**5.86**
0.13 ~ 0.14	42	12	**14.20**	-1.85	**11.73**	**57.75**	**20.46**
0.14 ~ 0.15	30	1	-1.85	**9.26**	**14.82**	-241.00	-54.69
0.15 ~ 0.16	29	7	**18.78**	**2.38**	**23.55**	-176.57	-32.96
0.16 ~ 0.17	22	4	**10.18**	**18.06**	0.00	**278.75**	**76.75**
0.17 ~ 0.18	18	2	**1.86**	-6.48	**18.52**	-215.50	-50.40
0.18 ~ 0.19	16	4	**276.90**	**270.42**	**282.00**	**644.00**	**368.33**
0.19 ~ 0.20	12	2	64.81	23.15	44.44	-570.50	-109.52



(그림 8) 통합 범주 분류 상에서 μ값의 구간별 분류 성능의 변화량



(그림 9) 기계학습 알고리즘의 분류 성능

자동화된 웹 페이지 분류 시스템을 구현하기 위해서 위에서 생성한 튜플들을 기계학습의 학습 데이터로 사용한다. 본 논문에서 사용한 기계학습 알고리즘은 C4.5[14], 베이지언 분류법(naïve Bayesian classifier)[4], CN2[2], 역전파(back- propagation) 신경망[24] 알고리즘으로 총 네 개이다. 이들 알고리즘에 의해 생성된 규칙들은 실시간으로 임의의 웹 페이지들을 분류하는데 사용된다.

최종 분류정확도를 측정하기 위하여, 위에서 생성한 규칙을 3,600개의 웹 페이지에 적용해 (그림 3)에서 제시한 네 개의 범주 중 하나로 분류하는 실험을 하였다. (그림 9)는 각각의 기계학습 알고리즘에 의해 생성된 분류 규칙이 갖는 분류 정확도 성능 측정값을 그래프로 나타낸 것이다. CN2의 성능은 89.05%로 네 개의 알고리즘 중에 가장 좋은 성능을 보여줬고, 가장 나쁜 성능을 보인 베이지언 분류법도 73.11%의 분류 성능을 보였다. 따라서 3,600개의 웹 페이지에 대해 실시한 분류 실험은 평균적으로 83.52%의 분류 성능을 가졌다고 할 수 있다.

6. 결론 및 향후과제

본 논문에서는 온톨로지에 기반한 자동화된 웹 페이지 분류 시스템을 제안하였다. 이러한 시스템을 개발하기 위하여, 온톨로지의 각 분류 범주에 속하는 단어들이 가져야 할 두 가지 속성을 (1) 각 분류 범주를 대표할 수 있으며, (2) 각 분류 범주를 다른 분류 범주와 뚜렷이 구분할 수 있는 특성으로 구체화 하였다. 따라서, 첫 번째 속성을 위하여 단어들의 가중치를 계산하여 각 분류 범주에 속하는 문서 집합을 대표하는 단어들을 선택하였으며, 두 번째 속성을 구현하기 위하여 선택된 단어들의 정보이득 값을 계산하여 우선순위를 결정하였다. 그리고 추출된 특성 즉, 이와 같이 선정된 단어들을 가지고 기계학습 알고리즘을 적용하여 분류 규칙을 생성하였다.

실험에서는 본 논문이 제안한 웹 페이지 분류 시스템의 분류 정확도에 대한 성능을 측정하였다. 그 결과 제안된 방법이 기계학습의 입력값으로 쓰이는 단어의 수를 효과적으로 줄일 수 있음을 알 수 있었다. 분류된 웹 페이지들의 집합으로부터 대표성을 갖는 단어를 추출하는 실험에서, 평균적으로 250개의 단어들로부터 분류 범주를 대표하는 78개의 단어를 선정할 수 있었다. 이때의 분류 정확도에 대한 성능은 약 83.52%를 보였다.

본 연구의 결과를 다양한 도메인의 웹 페이지 분류를 위하여 적용할 수 있다고 판단한다. 특성을 추출할 때 HTML 문서의 구조적 특징이나 하이퍼링크와 같은 웹 페이지의 특징을 부가하여 이 특징들에 의한 성능 향상을 실험해 볼 수 있으며, 여러 가지 다른 도메인의 웹 페이지 분류를 수행할 수 있을 것으로 기대한다. 또한 분류 시스템에 새로운 웹 페이지들이 지속적으로 입력될 때, 이에 의한 온톨로지와 분류규칙의 갱신 및 성능 변화를 실험해 볼 수 있을 것이다.

본 연구를 토대로 향후에는 다양한 도메인에 대한 웹 페이지 분류를 해볼 계획이며, 특히 자체적으로 수집한 30,000개의 웹 페이지들에 대하여 웹 페이지의 유해성 여부를 분류하여 시스템의 유용성을 분석할 계획이다. 또한 새로운 웹 페이지들을 이용하여 분류 시스템의 규칙을 갱신할 때, 본 논문에서 제안한 단어 선택 방법의 유용성을 검증해 볼 것이다. 즉, 제안된 방법을 사용하여 갱신된 분류 규칙이 새로운 웹 페이지의 특성을 어느 정도 반영하는지 분석하며, 갱신된 규칙을 기존의 웹 페이지 집합에 적용하여 성능을 평가할 계획이다.

참 고 문 헌

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press/Addison-Wesley, New York, 1999.

[2] P. Clark and T. Niblett, "The CN2 Induction algorithm," *Machine Learning Journal*, Vol.3, No.4, pp.261-283, 1989.

[3] C. Ding, C. Chi, J. Deng and C. Dong, "Centralized content-based Web filtering and blocking : how far can it go?," Proc. of 1999 IEEE International Conference on Systems, Man and Cybernetics, Vol.2, pp.115-119, October, 1999.

[4] R. Hanson, J. Stutz and P. Cheeseman, *Bayesian Classification Theory*, Technical Report FIA-90-12-7-01, NASA Ames research Center, AI Branch, 1991.

[5] L. Holder, ML v2.0, *Machine Learning Program Evaluator*, available on-line, http://ranger.uta.edu/~holder/courses/cse6363/ml2.0.tar.gz.

[6] C. Jenkins, M. Jackson, P. Burden and J. Wallis, "Automatic RDF metadata generation for resource discovery," Proc. of 8th International WWW Conference, Toronto, pp.11-14, May, 1999.

[7] Lawrence Berkeley National Labs Network Research Group, *libpcap*, available on-line, http://ftp.ee.lbl.gov.

[8] Y. Ng, J. Tang and M. Goodrich, "A binary-categorization

approach for classifying multiple-record Web documents using application ontologies and a probabilistic model," Proc. of 7th International Conference on Database Systems for Advanced Applications, pp.58-65, April, 2001.

[9] S. Noh, C. Lee, K. Choi and G. Jung, "Detecting Distributed Denial of Service(DDoS) Attacks Through Inductive Learning," Lecture Notes in Computer Science 2690, pp.286-295, Springer, 2003.

[10] S. Noh, H. Seo, J. Choi, K. Choi and G. Jung, "Classifying Web Pages Using Adaptive Ontology," Proc. of the IEEE International Conference on Systems, Man and Cybernetics, pp.2144-2149, Washington, D.C., October, 2003.

[11] N. F. Noy and D. L. Mcguinness, "Ontology development 101 : A guide to creating your first ontology," *Knowledge Systems Laboratory(KSL), Department of Computer Science, Stanford : Technical report*, KSL-01-05, 2001.

[12] S. Parent, B. Mobasher and S. Lytinen, "An adaptive agent for web exploration based on concept hierarchies," Proc. of 9th International Conference on Human Computer Interaction, New Orleans, August, 2001.

[13] R. Prabowo, M. Jackson, P. Burden and H. Knoell, "Ontology-Based Automatic Classification for the WEB Pages : Design, Implementation an Evaluation," Proc. of 3rd International Conference, Singapore, pp.182-191, 2002.

[14] J. R. Quinlan, *C4.5 : Programs for Machine Learning*, Morgan Kaufmann, 1993.

[15] J. Reynolds and J. Postel, "Assigned Numbers," STD 2, RFC 1700, October, 1994.

[16] D. Riboni, "Feature Selection for Web Page Classification," EURASIA-ICT 2002 Proc. of the Workshops, Shiraz, Iran, October 2002.

[17] S. M. Ruger and S. E. Gauch, *Feature Reduction for Document Clustering and Classification*, Technical report, Computing Department, Imperial College, London, 2000.

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

[19] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, Vol.34, No.1, pp.1-47, 2002.

[20] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, Vol.27, pp.379-423 and 623-656, July/October, 1948.

[21] M. P. Sinka and D. W. Corne, "A large benchmark dataset for web document clustering," *Soft Computing Systems : Design, Management and Applications, Frontiers in Artificial Intelligence and Applications*, Vol.87, pp.881-890, 2002.

[22] N. Soonthornphisaj, P. Chartbanchachai, T. Pratheeptham, and B. Kijsirikul, "Web page categorization using hierarchical headings structure," Proc. of 24th International Conference on Information Technology Interfaces, Vol.1, pp.37-42, 2002.

[23] A. Sun, E. Lim and W. Ng, "Web classification using support vector machine," WIDM'02, Virginia, November, 2002.

[24] D. R. Tveter, *Backprop Package*, available on-line, http://www.dontveter.com/nnsoft/bp042796.zip, 1996.

최 재 혁

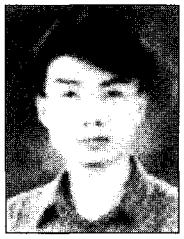
e-mail : jaehyuk04.choi@samsung.com

2002년 아주대학교 정보통신대학정보 및 컴퓨터공학부(학사)

2004년 아주대학교 정보통신전문대학원 정보통신공학과(석사)

2004년~현재 삼성전자 네트워크 사업부

관심분야 : 인터넷 보안, 인공지능, 실시간 시스템 등



서 혜 성

e-mail : retry@ajou.ac.kr

2003년 아주대학교 정보통신 대학 정보 및 컴퓨터공학부(학사)

2003년~현재 아주대학교 정보통신전문대학원 정보통신공학과

관심분야 : 네트워크 보안, 분산 시스템, 인공지능 등

노 상 욱

e-mail : sunoh@catholic.ac.kr

1987년 서강대학교 생명과학(이학사)

1989년 서강대학교 컴퓨터공학(공학석사)

1999년 텍사스 주립대(Arlington) 컴퓨터공학(공학박사)

1989년~1995년 국방과학연구소 연구원

2000년~2002년 미조리 주립대(Rolla) 컴퓨터학과 조교수

2002년~현재 가톨릭대학교 컴퓨터정보공학부 조교수

관심분야 : Knowledge Management, Intelligent Agent, Multi-Agent System, Machine Learning, Distributed Real-Time System 등

최 경 희

e-mail : khchoi@madang.ajou.ac.kr

1976년 서울대학교 사범대학 수학교육과 (학사)

1979년 프랑스 그랑데꼴 Enseeiht 정보공학과(공학석사)

1982년 프랑스 Paul Sabatier 정보공학과 (공학박사)

1982년~현재 아주대학교 정보통신전문대학원 교수

관심분야 : 운영체제, 분산시스템, 실시간 및 멀티미디어 시스템 등

정 기 현

e-mail : khchung@madang.ajou.ac.kr

1984년 서강대학교 공과대학 전자공학과 (학사)

1988년 미국 Illinois 주립대 EECS(공학석사)

1990년 미국 Perdue 전기전자공학부 (공학박사)

1991년~1992년 현대전자 반도체 연구소

1993년~현재 아주대학교 전자공학부 교수

관심분야 : 컴퓨터구조, VLSI설계, 멀티미디어 및 실시간 시스템 등