지지 벡터 기계를 이용한 계층적 문서 분류*

포항공과대학교 정보통신학과,[1] 컴퓨터공학과[2]
윤용욱[1†] · 이창기[2] · 이근배[2]

Hierarchical Text Categorization using Support Vector Machine

Yong Wook Yoon,[1] Changki Lee,[2] Gary Geunbae Lee[2]
Graduate School of Information Technology,[1] POSTECH, Pohang
Department of Computer Science and Engineering,[2] POSTECH, Pohang, Korea

요 약

인터넷을 통해 생성, 전달되는 문서 량이 급격히 많아짐에 따라, 정보의 접근을 용이하게 하기 위한 문서의 자동 분류 기능이 절실히 요구되고 있다. SVM(Support Vector Machine)은 최근에 문서 분류에 널리 쓰이고 있는 기법으로 다른 분류기에 비하여 좋은 성능을 보여주고 있다. 하지만 SVM은 현재까지 주로 비 계층 평탄화(flat)된 분류 응용에 효과적으로 적용되어 왔다. 이와 달리 본 논문은 문서 분류에 있어서 최종 분류 class를 한번에 출력하는 비 계층 분류 보다는, 비슷한 성질을 갖는 class의 집합을 계층적 구조로 묶어 분류하는 계층적 분류 기법이 보다 사람이 이해하기 쉽고 사용하기 편리하며 더 효과적이라는 것을 보이고, 실험을 통해 계층적 분류를 위한 효과적인 SVM 분류기를 개발하여 비 계층 분류보다 좋은 분류 성능을 보여 줄 수 있음을 확인한다.

서 론(1절)

사무실에서나 가정에서 PC 사용이 보편화되어 네트워크를 통한 문서의 생산 및 전달이 활발히 이루어 지고 있다. 또한 인터넷을 기반으로 한 전자출판의 발달로 매일 쏟아지는 문서 량이 급격히 증가함에 따라 이를 체계적으로 분류하여 이용할 수 있는 환경이 무엇보다도 중요해졌다. 문서의 자동분류 기술은 문서의 주제에 따라 사람이 아닌 기계가 자동으로 분류를 하여 주므로 실시간 인터넷 환경에서 더욱 필요한 기술이다. 문서 분류는 정보 검색의 한 분야로 일찍이 많은 연구가 진행되어 왔으며, 전반적인 내용과 과거의 연구 결과가 Sebastiani[1]에 체계적으로 잘 정리되어 있다.

문서가 속한 클래스 간에 어떤 계층적 구조가 있을 때 하나의 문서는 하나 이상의 클래스에 속하게 되고, 그 계층 구조의 상부로 올라갈수록 보다 광범위한 범주에 속하게 된다. 이러한 예는 웹 포탈 사이트의 디렉토리 서비스에서 쉽게 볼 수 있는데, 문서를 주제별로 계층을 만들어 디렉토리 형태로 분류하여 사용자들이 쉽게 찾을 수 있도록 한 것을 알 수 있다. 일반적으로 문서 분류시 클래스의 계층을 만들어 분류하는 것이 평탄화된 구조의 클래스 분류보다 보다 좋은 성능을 보여주고 있다.[2,3] 클래스의 계층구조를 형성하기 위해서는 먼저 유사한 클래스간의 군집을 형성해야 하는데, 사람이 직접 수작업으로 유사한 클래스를 묶을 수도 있고, 문서 량이 많아지거나 보다 정밀한 결과를 얻고자 할 때는 클러스터링 기법을 활용하기도 한다.[4]

문서분류에 있어서 문서의 표현방법은 정보검색에 널리 쓰이는 전통적인 Vector Space 모델을 많이 사용한다. 보통, 문서의 자동 분류를 위해서는 일단 모아진 문서집합에 대하여 분류기를 학습시키고, 학습된 분류기를 실제 문서에 대하여 실행시켜서 그 결과를 얻게 되는데, 지금까지 많은 형태의 분류기가 제안되었고 사용되고 있다.[1] 이러한 분류기들은 학습에 필요한 문서집합의 자질들을 추출하는

*본 연구는 산업자원부 중기거점과제 '차량정보센터용 대화체 음성 인식엔진 기술개발'의 지원을 받아 수행되었음.
†E-mail : ywyoon@postech.ac.kr
E-mail : leeck@postech.ac.kr
E-mail : gblee@postech.ac.kr

과정은 동일하나 이를 학습하고 클래스를 예측하는 방법에 있어서는 분류기마다 차이를 보이고 있다.

최근 들어 문서분류 분야에서는 Vapnik[5]이 제안한 지지 벡터 기계(Support Vector Machine, SVM) 분류기가 각광을 받고 있다. SVM 분류기는 많은 양의 데이터와 높은 차원의 자질집합을 가진 분류작업에 특히 우수한 성능을 보인다고 알려져 있다. 따라서 현대의 대량 문서의 자동분류 작업에 적합하다고 하겠다. 하지만 본 저자의 지식으로는, SVM은 여태까지 평탄화(flat)된 비 계층 문서 분류에 주로 적용되어 최고의 성능을 보여 왔다. 본 논문에서는 이런 SVM의 성능을 문서의 계층적 분류를 위해 효과적으로 활용하는 방법을 제안한다.

본 논문은 다음과 같은 구조를 가진다. 2절에서는 문서의 계층적 분류방법의 특성에 대해서, 3절에서는 SVM을 이용한 문서분류와 그 계층적 활용에 대해서 각각 알아본다. 그리고, 4절에서는 SVM을 이용한 계층적 문서분류 구현 및 실험에 대한 결과와 분석을 싣고, 5절에서는 결론 및 향후 연구 과제에 대하여 논한다.

문서의 계층적 분류(2절)

문서 클래스의 집합을 C라 하고 각각의 클래스를 c_1, c_2, ..., c_n이라고 하자($|C|=n$). 문서 분류를 위한 클래스 구성방법 두 가지를 Fig. 1에 나타내었다. 먼저 (a)는 모든 클래스를 루트 노드 아래에 평탄화(flat)된 구조로 배치한 것이고, (b)는 클래스를 적당한 크기의 부분 집합으로 묶고 그 부분집합들을 계층적 구조로 배치한 것이다. 문서분류 방법에 있어 (a)와 같은 경우를 평탄화된 분류방법(Flat Classification method), (b)를 계층적 분류 방법(Hierarchical Classification method)이라고 부른다.

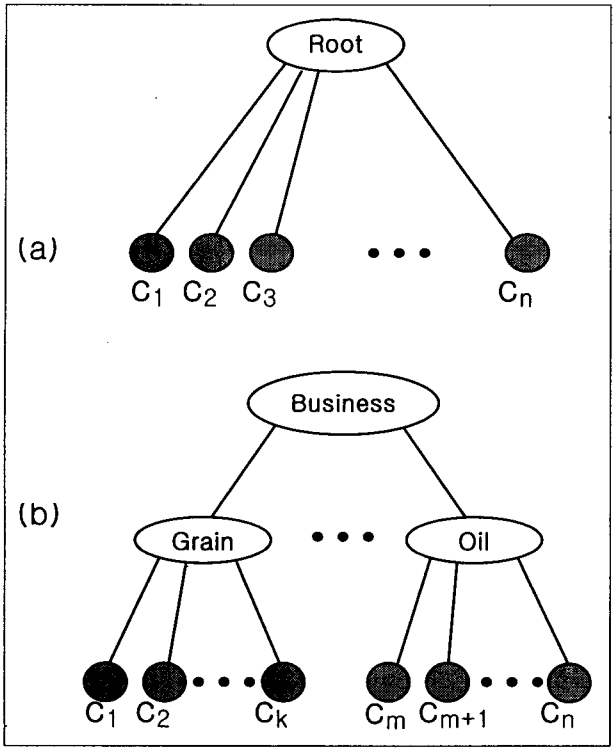

Fig. 1. 문서 클래스의 구성 방법.

클래스가 계층적으로 구성되었을 때 한 문서는 여러 클래스에 속하는 것이 원칙이다. 이러한 계층적 문서분류에서는 한 문서가 루트 노드에 해당하는 가장 광범위한 클래스에서부터 아래로 내려가면서 가장 세부적인 클래스까지 차례로 분류가 이루어지게 된다.

계층적 문서 분류 방법은 문서 량이 대단히 많아 질 때 더욱 유용해 지며, 사람의 직관적인 이해와도 가깝기 때문에 실제 환경에 사용하는데 적합하다. 웹 상의 문서 검색에서는 이런 분류 방법이 필수적이며 많은 연구가 행하여지고 있다.[6]

계층적 분류 방법은 분류가 단계별로 이루어 지므로 일반적인 방법보다 분류기 수가 더 많은 것이 보통이다. Fig. 1에서 알 수 있는 것처럼, 일반적인 방법에서는 루트 노드에서만 분류기가 존재하지만, 계층적 방법에서는 매 단계마다 다른 분류기가 있어야 한다. (b)에서는 루트 노드의 문서집합을 "Business"라 하고, 그 하부 클래스를 "Grain", "Oil" 등으로 나누어서 그에 해당하는 클래스에 대한 분류작업을 담당하며, "Grain" 노드에서는 마찬가지로 해당 하부 노드 클래스들에 대한 분류 작업을 담당하는 것이다.

미리 계층구조가 정해져 있다고 가정하면, 계층적 문서분류는 각각의 노드마다 분류기를 할당함으로써 가능해진다. 어떤 분류기를 할당 하느냐에 따라 두 가지 다른 전략이 존재한다.[7]

1) 각각의 non-leaf 노드 마다 m-way 분류기[1]를 할당
2) 각각의 부모-자식 노드 쌍마다 이진 분류기를 할당

Q_m, Q_2를 각각 1), 2)의 학습에 있어서의 계산 복잡도라 하면 다음과 같다.[7]

$$Q_m = \sum_{i=0}^{h-1}\sum_{j=1}^{m_i} O(n_{ij}^c),\ Q_2 = b \times \sum_{i=0}^{h-1}\sum_{j=1}^{m_i} O(n_{ij}^c) \quad (1)$$

여기서,
h는 계층의 깊이,
m_i는 i번째 레벨의 클래스 수

1 분류 대상 클래스의 수가 3 이상인 분류기

i=0, 1, 2, ..., h, i=0은 루트 노드에 해당,

j=1, 2, 3, ..., m_i는 레벨 i에 있어 클래스 수 순위,

n_{ij}는 학습대상 문서 수,

b는 평균 branching factor이다.

즉, 학습에 소요되는 시간은 매 단계 학습 대상 문서수의 c 제곱에 비례하며, 총 학습 시간은 그것들을 다 합한 것과 같다.

계층적 문서분류의 장점은 일반적 방법에 비해 보다 유연하고 정교한 분류 전략을 선택할 수 있다는 것이다. 각 클래스 집합은 문서의 수나 문서의 성격에 차이가 있으므로 이를 고려하여 거기에 최적화된 학습 모델을 구하면 효율(Efficiency)이나 성능(Effectiveness)의 극적인 향상을 도모할 수 있다. 예를 들면 각 노드 마다 다른 분류기를 선택한다든지, 같은 분류기일 경우 서로 다른 파라미터를 적용하는 것 등을 들 수 있다.

지지 벡터 기계를 이용한 문서 분류(3절)

1. 지지 벡터 기계

지지 벡터 기계(Support Vector Machine, SVM)은 '구조적 위험도 최소화'를 목표로 하는 범용적인 통계적 학습 체계이다.[5] l개의 학습 표본 $\{(x_1, y_l), ..., (x_l, y_l)\}$이 주어졌다고 할 때, SVM의 학습은 비선형 최적화 문제로 귀결되며, 학습 결과 아래와 같은 결정 함수를 얻는다.

$$f(\mathbf{x}) = \mathrm{sgn}\left[\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x} \cdot \mathbf{x}_i) + b\right] \qquad (2)$$

여기서 αi는 각 학습표본마다 주어지는 가중치 값으로 0이 아닌 경우의 x_i를 support vector라고 하며 클래스를 결정하는데 있어서 필수적인 정보를 담고 있다. (2)에서 알 수 있는 것처럼 SVM은 기본적으로 이진(binary) 결정 함수를 제공한다. 예를 들어, x라고 하는 문서가 들어 왔을 경우, (2)식에 따라 부호를 계산하여 양이면 해당 클래스, 음이면 그 클래스에 속하지 않는다고 판별하는 것이다. 그 값을 구하기 위해서는 우선 $K(x \cdot x_i)$를 계산해야 하는데, 이를 커널 함수라 한다. 커널 함수엔 RBF, Polynomial 등 여러 가지가 있으나 문서분류에는 일반적으로 선형(Linear) 커널 함수가 사용이 간편하고 성능도 우수하다고 알려져 있다.[8,4]

2. 단일 레이블 분류와 다중 레이블 분류

문서 분류 방법은 또한 단일 레이블(Single-Labeled) 분류와 다중 레이블(Multi-Labeled) 분류로 나눌 수 있다.[1] 클래스 집합을 C= $\{c_1, c_2, ..., c_n\}$라 하고 문서집합을 D라 하면, 모든 문서에 $d \in D$대하여 단 하나의 클래스 c_i로 분류되는 경우를 단일 레이블 분류라 하고, 문서 d에 대하여 0에서 |C|까지 수의 클래스가 분류 될 수 있다고 하면 그 경우를 다중 레이블 분류라 한다. 실 세계에 존재하는 문서의 경우 다중 레이블인 경우가 많으며, 앞 절의 계층적 분류에서 하나의 문서가 여러 클래스에 속하는 경우와는 좀 다른 관점의 문제이다(동일 레벨에서의 분류에 대해서 초점을 맞추는 것이다).

SVM은 기본적으로 이진 분류 기능만을 제공하지만 클래스수가 m일 경우 m개의 독립적인 SVM을 가지고 다중 레이블 분류를 행할 수가 있다. 이 경우 분류의 Effectiveness Measure로서 정확률(Precision), 재현율(Recall)을 다음과 같이 정의 할 수 있다. 임의의 클래스 c_i에 대하여 그에 해당하는 이진 분류기 h_i의 분류결과를 아래와 같이 contingency table로 표시하자.

Table 1에서 TP는 True Positive에 해당하며 올바로 분류된 경우의 수의 합이다. FP, FN은 각각 False Positive, False Negative의 약자이다. 분류기 h_i의 클래스 c_i에 대한 정확률 P_i와 재현율 R_i는 각각 다음과 같다.

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i} \qquad (3)$$

다중 레이블 분류를 고려하여 m개의 클래스에 대한 m개의 SVM으로 구성된 분류기의 전체적인 성능은 보통 micro-averaged Breakeven Point로 나타내며, 전체 Precision, Recall을 P, R이라고 할 경우(P+R)/2로 계산된다.[8]

분류기의 성능을 나타내는 다른 척도로서 Accuracy가 있다. SVM의 경우 결정함수 값의 음양에 상관없이 가장 큰 점수를 획득한 클래스가 정답 클래스에 속할 경우 올바로 분류되었다고 간주하고 그 경우의 수를 전체 문서의 수로 나눈 값이 Accuracy가 된다. 일반적으로 Accuracy measure는 Contingency Table상의 수치 중 outlier에 민감한 특성을 나타내므로 분류기의 객관적인 성능을 나타내기에는 미흡한 점이 있다.[9]

Table 1. 클래스 c_i에 대한 Contingency Table

클래스 c_i		전문가 판정	
		Yes	No
분류기 판정	Yes	TP_i	FP_i
	No	FN_i	TN_i

3. 계층 분류와 SVM의 결합

문서 집합의 클래스들 간의 분포를 잘 반영한 계층적 분류방법과 문서 분류에 있어서 높은 성능을 보이는 SVM 분류기를 결합하여 자동 문서분류 작업에 적용하면 더욱 탄력적이며 높은 성능을 얻을 수 있다. 더욱이 웹과 같이 대량의 문서가 유통되는 곳에선 문서의 계층이 기하급수적으로 늘어 날 수 있으며 여기에서도 SVM은 계층적 분류와 결합하여 이상적인 조합을 나타낸다. [7]에서 Yang 등은 계층적 분류에서의 이론적인 계산복잡도를 여러 종류의 분류기에 대하여 제시하고, 실험을 통하여 레벨 깊이가 10, 평균 branching factor 9인 계층적 문서 분류 결과를 통해 SVM이 학습시간과 판정시간 측면에서 k-NN 분류기를 사용했을 경우 보다 월등한 결과를 보여주고 있다.

계층적 SVM을 위한 구현 및 실험결과 분석(4절)

실험을 위한 문서 집합으로 20 Newsgroups 문서집합을 사용하였다.[10] SVM을 학습시키기 위한 전 단계로 문서들의 자질 집합을 추출하고 가공하는 데는 McCallum [11]의 'Bow' Toolkit를 사용하였다. SVM 분류기는 구현된 것이 많이 있는데, 본 실험에서는 'Bow' Toolkit에서 제공되는 SVM을 이용한 결과만을 수록하였다.

Xref: cantaloupe.srv.cs.cmu.edu talk.religion.misc:82764 alt.atheism:51138
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!husc-news.harvard.edu!hsdndev!wupost!zaphod.mps.ohio-state.edu!pitt.edu!pogo.isp.pitt.edu!joslin
Newsgroups: talk.religion.misc,alt.atheism
Subject: Re: *** The list of Biblical contradictions
Message-ID: <7912@blue.cis.pitt.edu>
From: joslin@pogo.isp.pitt.edu(David Joslin)
Date: 5 Apr 93 20:41:08 GMT
Sender: news+@pitt.edu
Followup-To: talk.religion.misc
References: <bskendigC50tnu.lno@netcom.com>
Organization: Intelligent Systems Program
Lines: 44

Someone writes:
>I found a list of Biblical contradictions and cleaned it up a bit, but
>now I'd like some help with it.

I'm curious to know what purpose people think these lists serve. Lists like this seem to value quantity over quality, an "argument from article length." And the list you have here is of poorer quality than most. Since the quotes seem to be taken from an on-line bible, I doubt that there will be much problem with verses quoted inaccurately. But that isn't the problem here.

Fig. 2. 'alt.atheism' newsgroup의 기사(일부).

1. 20 Newsgroups 문서집합

20 Newsgroups 문서집합은 USENET의 20개 newsgroup에 기고된 기사들로 구성되었으며, 일찍이 문서분류 실험에 널리 사용되어 왔다. USENET에는 수많은 Newsgroup이 존재하며 따라서 그 자체로서 계층구조를 자연스럽게 형성하고 있다. 아래 Fig. 2는 기사의 한 예를 보여주고 있다.

기사는 상단부의 Header부분과 하단의 Body부분으로 크게 나뉘어져 있으며, Header부분에는 이 기사가 속한 newsgroup, 제목, 보낸이 등이 표시돼 있다. 보통 실험에 사용할 때는 Header부분을 삭제하고 Body만을 가지고 사용한다. 본 실험에서는 Header부분 중 'Subject' 필드와 Body부분을 사용하였다.

또한, 기사는 둘 이상의 newsgroup에 속하는 경우가 있는데(여기서는 하나의 newsgroup이 하나의 class에 해당), 이 경우를 고려하면 다중 레이블 분류에 해당한다. Fig. 2의 기사의 경우 Header부분을 보면 'alt.atheism'과 'talk.religion.misc' 두 클래스에 속함을 알 수 있다. 총 19,997개의 문서 중 504개의 문서가 둘 이상의 클래스에 속하여 있다.

본 실험에서는 아래 Table 2처럼 본래 속한 newsgroup을 기반으로 계층 구조를 형성하였다.

위 Table의 각 셀의 class마다 해당 이진 분류기를 구성하면 2 절의 계층적 분류기 구성 전략 2)를 따르는 것이 된다. 이 구성에서는 Level 0에서 leaf 노드에 해당하는 분류기 3개, 중간 노드에 해당하는 분류기 4개를 구성하게

Table 2. 20 Newsgroups의 계층 구조

Level 0	Level 1	Level 2
alt.atheism misc.forsale soc.religion.christian		
comp	graphics os.ms-windows.misc windows.x	
	sys	ibm.hardware mac.hardware
rec	autos motor-cycles	
	sport	baseball hocky
sci	crypt electronics med space	
talk	religion.misc	
	politics	guns mideast misc

되며, Level 1, 2에서도 마찬가지 방법으로 분류기를 구성한다. 그러면 총 27개의 2진 분류기가 되는데 평탄화된 분류방법과 비교하면 7개의 중간 노드 분류기가 더 소요 됨을 알 수 있다.

Table 2의 계층구조는 20 Newsgroups의 표층적인 구조만을 고려한 것으로서 분류성능을 고려한 최적의 구조는 아닐 것이다. 최적의 결과를 얻기 위해선 그룹 내 문서들의 자질들에 대한 분석이 선행되어야 하며, [4]에서 제시한 클러스터링 방법도 하나의 대안이 될 수 있다.

2. 계층적 SVM 구현 방법

실험은 평탄화된 구조의 분류 성능과의 비교를 위하여 평탄화 분류와 계층적 분류를 병행하였으며, 모두 SVM분류기와 결합하여 구현되었다.

20 Newsgroups 전체 문서에 대하여 4-fold cross validation을 행하였으며, 불용어(stop-word)는 제외하고, stemming은 하지 않았으며, 본문 내용이 binary인 문서는 제외시켰다. SVM의 커널 함수로는 선형 함수를 사용하였으며, Cost 파라미터에 대한 최적화 작업을 분류기마다 독립적으로 시행하였다.

Fig. 3에 계층적 SVM 문서분류 구현에 있어서 학습 단계에 해당하는 절차를 나타내었다. 먼저 전체 문서 집합으로부터 분류하고자 하는 클래스를 결정한 다음 그 클래스 분류를 위한 최적의 계층구조를 논리적으로 설정한다. 이를 수작업으로 할 수도 있고 특화된 클러스터링 알고리즘이 사용될 수도 있다. 다음으로 설정된 계층구조에 따라 분류기를 선정하고 그 분류기에 해당하는 학습대상 문서를 원 문서 집합에서 분리하여 각 분류기에 배정한다. 그리고 나서, 각 분류기마다 학습문서를 가지고 학습을 행하여 최종적으로 학습된 모델을 생성한다. 각 단계별 최적화를 위하여 cross-validation을 활용한다.

학습이 끝나면 판정 대상 문서집합을 준비한 후 학습된 분류기에 차례로 입력하여 실행시킨다. 분류는 루트 노드에서부터 시작하여 Pachinko-Machine 형태로 Yes로 판정된 노드의 하위 분류기로 내려가게 된다(Fig. 4).

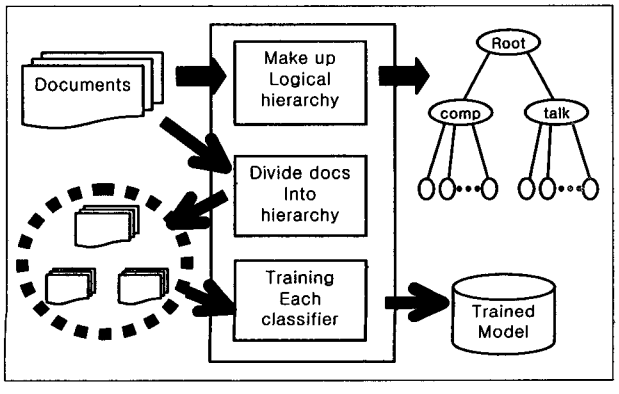

Fig. 3. 계층적 SVM 문서분류-학습 단계.

Fig. 4에서 둥근 노드는 분류기를, 사각 노드는 분류가 끝나 최종 판정을 받은 문서가 담기는 노드를 가리킨다. 다중 레이블 분류이므로 판정 대상 문서가 2개의 클래스를 가졌을 경우, 동시에 2개의 하위 분류기로 문서가 내려가는 경우가 발생 할 수 있다. 또한 극단적인 경우, 해당 레벨의 모든 이진 분류기로부터 음의 판정결과를 받아 어느 클래스에도 속하지 않는 경우도 발생할 수 있다.

이같이 문서 분류 시스템이 여러 개의 이진 분류기의 결과에 좌우되므로 전체적인 성능 평가를 위해 3.2절 (3)의 단위 분류기의 성능을 토대로 전반적인 정확률(P)과 재현율(R)을 다음과 같이 산출한다.

$$P = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} TP_i + \sum_{i=1}^{m} FP_i}, R = \frac{\sum_{i=1}^{m} TP_i}{\sum_{i=1}^{m} TP_i + \sum_{i=1}^{m} FN_i}$$

여기서 m은 전체 클래스의 수이다. 총괄적인 성능을 나타내는 단일 지표로는 Breakeven Point, F-measure 등이 있으며, 이진 분류의 경우 micro-averaged Breakeven Point는(P+R)/2로 계산된다. 본 실험에서는 micro-averaged Breakeven Point를 사용하였다.

3. 실험 결과 및 분석

아래 Table 3에 평탄화 방법과 계층적 방법을 사용하여 얻은 결과를 나타내었다.

일반적 방법(Flat method)에서 Recall이 낮은 현상은 다

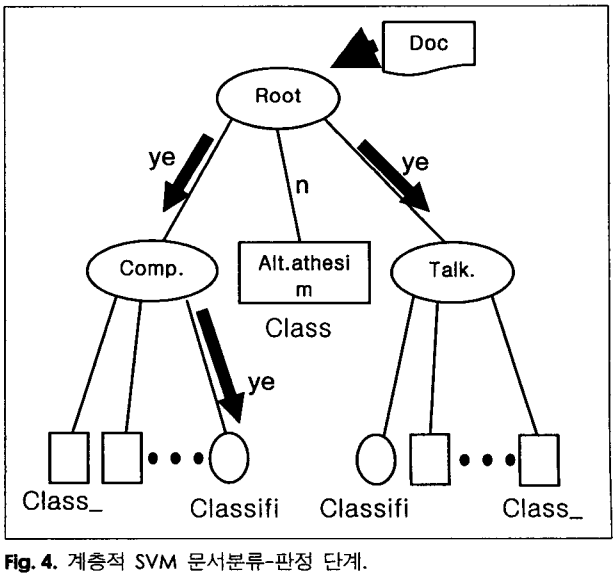

Fig. 4. 계층적 SVM 문서분류-판정 단계.

Table 3. 평탄화 방법과 계층적 분류의 비교-1

	Precision	Recall	Micro-BEP
Flat	96.73	40.37	68.55 (88.6)
Hierarchical	95.16	88.75	**91.55**

Table 4. 일반적 방법과 계층적 분류의 비교-2

	Accuracy	Accuracy (Previous results)
Flat	93.42	91.0(12)
Hierarchical	**94.34**	96.3(4)

중 레이블 문서의 분류시 첫번째 클래스에서는 정답을 내었으나 두번째 클래스는 거의 정답을 내지 못한 것에 기인한다. 이는 학습대상 자료를 만들 때 첫 번째 레이블이 정답인 것만 포함되어 두 번째 정답에 대한 Recall이 낮아지게 된 결과이다. 보다 정밀한 실험을 위해서는 다중 레이블을 포함한 모든 문서에 대해 학습자료를 추가로 만들어 학습시켜야 될 것이다. 그러나, 계층적 방법에서는 이를 극복하여 높은 재현율을 보여줌을 알 수 있다. 이는 계층구조 형성시 다중 레이블이 속한 클래스가 자연스럽게 클러스터링되면서 두 번째 레이블이 병합되어 Recall이 높아진 결과이다. 시스템의 전반적인 성능을 나타내는 Micro-BEP 값 91.55는 최근에 가장 좋은 성능을 보인 [8]의 88.6 보다 높음을 알 수 있다([8]에서는 평탄화 분류를 사용했음).

Table 4에서는 전통적 척도인 Accuracy를 나타내었다. 계층적 방법이 일반적 방법보다 높음을 알 수 있다. 그 오른쪽 열의 수치는 다른 실험에서 행해진 것들이다. 먼저 평탄화 분류 방법의 수치는 Probabilistic TF/IDF 분류기를 사용한 결과이다.[12] 계층적 분류의 경우 [4]에서의 실험 결과 값으로서, 다중 레이블 분류 결과도 아닌데 수치가 굉장히 높다. 이는 자질 집합을 추출할 때 기사 Header 부분의 'Organization' 필드를 포함한 결과이다. 저자가 행한 별도의 단일 레이블 실험에서 그 필드를 포함해서 학습을 시킨 결과, 그렇지 않은 경우 보다 10%정도 Accuracy의 향상을 가져왔으므로 본 실험 결과와의 직접적 비교는 무의미하다고 할 수 있다.

위에서 살펴본 바와 같이 SVM을 이용한 본 연구의 계층적 문서 분류방법이 과거 다른 실험 결과보다, 그리고 평탄화된 분류 방법보다 성능이 좋은 이유는 개개의 분류 단계마다 그에 최적을 이루는 분류기를 각각 학습시킬 수 있다는 데 있다. 본 실험에서는 매 분류단계 마다 SVM 분류기를 사용하였지만 해당 문서집합의 성격에 따라 그에 맞는 다른 분류기의 선정이 가능한 것이다. 이렇듯 분류기의 성능 향상에 있어 유연성이 있음을 알 수 있다. 반면, 실행 시간에 있어서는 계층적 분류방법이 추가의 중간 노드 분류기를 학습시키는데 소요되는 학습시간은 무시할 만 하다. 왜냐하면, 평탄화된 방법을 사용하여 학습을 할 경우에는 전체 문서 집합에 대하여 학습을 하므로, 그 문서수의 1.5~2 제곱에 비례하여 학습시간이 소요되므로 부분 집합으로 나뉘어 학습시키는 계층적 분류 방법에 비해 학습시간이 더 소요됨을 알 수 있다. 이렇듯 SVM은 계층적 분류방법과 결합하여 훌륭한 시너지 효과를 나타냄을 알 수 있다.

결론 및 향후 과제(5절)

문서의 자동 분류에 있어 문서량이 많아 질수록 계층적 분류 방법이 필요하며 성능이나 학습시간 측면에서도 계층적 분류 방법이 일반적인 경우보다 월등 함을 알 수 있다. 특히 SVM 분류기는 문서 분류에 좋은 성능을 나타내는 것에 더하여, 계층적 분류방법과 결합하여 사용되어질 경우 지금까지 그 어떤 문서분류 방법보다 뛰어난 성능을 보여준다.

향후, 계층적 분류를 하기 위한 계층적 구조를 수작업으로 하지 않고, 문서 집합의 성질을 고려한 자동화된 클러스터링을 통한 계층 구조의 구축이 요구되며 이를 위한 실험을 계획하고 있다. 아울러 실 세계에서의 응용성을 높이기 위하여, 문서가 점진적으로 추가되는 온라인 환경하에서의 학습방법과 실시간 판정 성능의 향상을 위한 연구도 필요하다고 생각한다.

REFERENCES

1) Fabrizio Sebstiani(2002) : *"Machine Learning in Automated Text Categorization", ACM Computing Surverys, Vol.34, No.1, March, pp1-47*
2) Koller D, Sahami M(1997) : *"Hierarchically classifying documents using very few words", Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97), pp170-178*
3) Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, Andrew Y. Ng (1988) : *"Improving Text Classification by Shrinkage in a Hierarchy of Classes", Proceedings of ICML-98, 15th International Conference on Machine Learning, pp359-367*
4) Tao Li, Shenghuo Zho, Mitsunori Orkhara(2003) : *"Topic Hierarchy Generation via Linear Discriminant Projection", Proceedings of SIGIR 2003, the Twenty-Sixth Annual International ACM SIGIR Conference, pp421-422*
5) Vapnik V(1995) : *"The Nature of Statistical Learning Theory", SpringerVerlag*
6) Dumais ST, Chen H(2000) : *"Hierarchical classification of Web content", In Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval (Athens, Greece, 2000), pp256-263*
7) Yiming Yang, Jian Zhang, Bryan Kisiel(2003) : *"A Scalability Analysis of Classifiers in Text Categorization", In Proceedings of SIGIR-03, 26th ACM International Conference (Toronto, Canada, 2003), pp96-103*
8) Bekkerman R, El-Yaniv R, Tkshby N, Winter Y(2001) : *"On Fea-*

ture Distributioanl Clustering for Text Categorization", Proceedings of SIGIR 2001, the Twenty-Fourth Annual International ACM SIGIR Conference, pp146-153

9) Lewis DD(1992) : *"An evaluation of phrasal and clustered representations on a text categorization task." In Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval (Copenhagen, Denmark, 1992), pp37-50*

10) Lang K(1995) : *"NEWSWEEDER : learning to filter netnews." In Proceedings of ICML-95, 12th International Conference on Machine Learning (Lake Tahoe, CA, 1995), pp331-339*

11) McCallum, Andrew Kachites : *"Bow : A toolkit for statistical language modeling, text retrieval, classification and clustering." http://www.cs.cmu.edu/~mccallum/bow*

12) Joachims T(1997) : *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Proceedings of ICML-97, 14th International Conference on Machine Learning (Nashville, TN, 1997), pp143-151*