복합 분류기를 이용한 웹 문서 범주화에 관한 실험적 연구

An Experimental Study on Categorization of Web Documents Using an Ensemble Classifier

이혜원, 정영미, 연세대학교 대학원 문헌정보학과

Hye-Won Lee, Young-Mee Chung
Dept. of Library and Information Science, Graduate School of Yonsei University

본 연구에서는 웹 문서를 분류하기 위해 문서로부터 다양한 자질을 추출하고, 두 가지의 분류기를 통해 여러 개의 분류 예측치를 구한 다음, 그것들을 하나의 결과물로 통합하는 복합 분류기를 사용하였다. 먼저 다양한 자질 집합에 대해 일반적으로 많이 사용되는 kNN(k nearest neighbor) 분류기와 나이브 베이즈(Naive Bayes) 분류기를 사용한 범주화 실험을 수행하고, 실험을 통해 나온 범주 예측치를 통합하는 복합 분류기들의 성능을 비교하였다. 또한 단일 분류기들을 통해 나온 모든 범주 예측치를 통합하는 과정을 수행하여, 단일 분류기만을 사용할 경우와 복합 분류기를 사용할 경우를 비교해 더 좋은 성능을 나타내는 분류기를 밝히고자 한다.

1 서 론

웹 문서를 범주화하는 기법은 기본적으로 학습예제를 이용하여 새로 입력되는 문서에 주제를 할당한다는 점에서 일반 문서를 범주화하는 방법과 유사하다고 할 수 있다. 그러나 웹 문서가 하이퍼텍스트라는 특성을 고려할 때 하이퍼링크된 문서의 정보를 이용하면 좀더 많은 자질을 분류 기준으로 삼을 수 있을 것이다.

웹 문서 범주화에서 단어는 그 위치에 따라 다른 유형의 자질로 사용될 수 있으며, 다양한 자질을 이용하는 것은 범주화의 성능을 높이기 위한 하나의 방법으로 볼 수 있다. 최근 들어, 상이한 데이터 셋을 포함하는 실험문서 집단을 범주화하는 실험에서 복합 분류기(ensemble classifier)를 사용하는 경우가 단일 분류기를 사용할 때보다 분류 성능이 향상되었음이 보고되고 있다(Dietterich 2000; Kuncheva, Skurichina, and Duin 2002).

본 연구에서는 웹 문서를 분류하기 위해 문서로부터 다양한 자질을 추출하고, kNN과 나이브 베이즈 분류기를 통해 여러 개

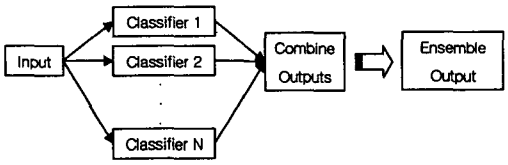
의 분류 예측치를 구한 다음, 그것들을 하나의 결과물로 통합하는 복합 분류기를 사용하였다.

2 복합 분류기에 의한 문서 범주화

복합 분류기는 다양한 분류기 또는 분류 작업을 통해 나온 범주 예측치들을 배깅(bagging)과 부스팅(boosting) 등의 방법에 의해 하나의 결과물로 통합하는 새로운 분류 기법을 의미한다(kuncheva, Skruichina, and Duin 2002).

배깅과 부스팅은 학습문서 집합을 무작위로 샘플링하는 기법을 사용함으로써 각 단일 분류기가 각기 다른 학습문서 집합을 가지고 학습한 결과물들을 결합한다(Opitz, and Maclin 1999).

〈그림 1〉은 복합 분류기의 개념을 도식화한 것이다.



〈그림 1〉 복합 분류기 개념도

복합 분류기는 문서 범주화에서도 단일 분류기만을 사용하였을 때보다 더 나은 성능을 보여주고 있다(Craven, Slattery 2001; Furnkranz 2002; kuncheva, Skruichina, and Duin 2002).

2.1 배깅(bagging)

배깅(bagging : Bootstrap AGGregatING)은 Breiman(1996)이 제안한 부트스트랩 샘플에 기반한 복합 분류기법이다. 배깅 기법에서 각 분류기의 학습문서 집합은 N개의 예제로 구성되는 원래의 학습 집합으로부터 무작위로 N개의 예제를 추출함으로써 생성된다(Opitz, and Maclin 1999).

〈그림 2〉는 배깅 기법의 예시로서 8개의 예제로 구성된 학습문서 집합으로부터 복원 추출된 4개의 학습문서 집합을 보여준다.

배깅 기법을 사용한 분류 과정은 다음과 같다(Ledward 1999).

① $C(x, t)$는 입력값 t에서 출력값을 k-vector로 가지는 분류기라고 가정한다. 여기서 $x=\{x_1, x_2, \cdots, x_n\}$으로 분석용 자료이고, $x_i=(t_i, y_i)$를 의미하며 t_i는 입력값, y_i는 출력값을 나타낸다.

② 복원 추출(replacement sampling)을 통해서 크기가 B인 부트스트랩 샘플 x^1, x^2,… x^B 을 생성하여 원래의 데이터 집합을 대체하고, 각 분류기를 학습시킨다.

③ 아래의 식에 의해 가장 큰 값을 가지는 y에 분류하게 된다.

$$C_{bag}(t)=\frac{1}{B}\sum_{b=1}^{B} C(x^b, t)$$

단일 분류기
학습 집합(training set)
training-set-1 : 1,2,3,4,5,6,7,8

⇩

배깅 기법에 기반을 둔 복합 분류기
복원 추출된 학습 집합(resampled training set)
training-set-1 : 2,7,8,3,7,6,3,1 training-set-2 : 7,8,5,6,4,2,7,1 training-set-3 : 3,6,2,7,5,6,2,2 training-set-4 : 4,5,1,4,6,4,3,8

〈그림 2〉 배깅 기법에 기반을 둔 복합 분류기의 학습문서 집합의 예

이러한 배깅 기법은 결정트리(decision tree) 기반 분류기와 같은 불안정한 분류과정의 편차를 현저히 줄여주며, 결과적으로 정확성 향상을 기할 수 있게 해 준다.

2.2 부스팅(boosting)

부스팅(boosting)은 Freund와 Schapire (1996)에 의해 새롭게 정의되어지고 Arcing (Adaptive Resampling and CombinING)과 AdaBoosting (Adaptive bootstrapping) 방법을 포함한다. 부스팅의 기본 기법은 배깅과 비슷하나 좀 더 독립적이고 무작위 추출에 적합하도록 되어 있다. 부스팅은 샘플링 할 때의 확률 분포가 관측치마다 다르기 때문에 서로 다른 확률분포를 가지고 다시 부트스트랩 샘플링을 한다는 것이다. 즉 오분류 되는 예측치에 더 큰 가중치를 줌으로써 분류기의 성능을 높이고자 하였다. 즉 부스팅에서는 N개의 예제로 된 학습 집합에서 N개의 새로운 학습 집합을 무작위로 추출한다. 처음에 추출된 학습 집합을 기준으로 오분류된 예제들의 수를 늘리면서 학습 집합을 새롭게 구축한다.

〈그림 3〉은 부스팅에서 8개의 예제로 구성된 학습문서 집합으로부터 복원 추출된 4개의 학습문서 집합을 보여주고 있다. 처음으로 복원 추출된 학습문서 집합(training- set-1)은 배깅의 것과 동일하게 출발한다. 하지만 학습이 진행되는 과정에서 오분류된 예제(1)의 수를 늘리면서 학습집합을 재구성한다.

부스팅 기법에 기반을 둔 복합 분류기
복원 추출된 학습문서 집합(resampled training set)
training-set-1 : 2,7,8,3,7,6,3,1 training-set-2 : 1,4,5,4,1,5,6,4 training-set-3 : 7,1,5,8,1,8,1,4 training-set-4 : 1,1,6,1,1,3,1,5

〈그림 3〉 부스팅 기법에 기반을 둔 복합 분류기의 학습문서 집합의 예

배깅은 부스팅에 비해 다소 낮은 정확률을 보이기도 하지만, 언제나 단일 분류기를 사용했을 때보다는 성능이 좋은 것으로 판명되었다. 그러나 부스팅의 경우에는 단일 분류기를 사용했을 때보다 더 낮은 성능을 보일 때도 있었다(Opitz, and Maclin 1999).

본 연구에서는 단일 분류기와 복합 분류기의 성능을 비교 평가하고자 함으로, 단일 분류기에 비해 언제나 성능이 좋게 나타나는 배깅의 기법을 기반으로 한 복합 분류

기를 활용하였다. 여러 개의 범주 예측치를 하나의 주제 범주로 결정하기 위해 배깅 기법 중 단순한 투표 방식, 가중치의 합을 구해 가장 최고값을 갖는 범주에 할당하는 가중치합 방식, 가장 높은 가중치를 가지는 하나의 범주에 할당하는 최대 신뢰도 방식 등을 이용하였다.

3 웹 문서 범주화 실험

3.1 실험 집단

범주화 실험의 대상이 되는 웹 문서를 수집하기 위해 yahoo(www.yahoo.com)의 디렉토리 서비스를 이용하여 미리 분류되어 있는 웹문서를 살펴보았다. 대상 웹 문서 추출 기준은 각 주제의 문서 규모의 수가 180~250개 정도로 되어 있는 주제를 선정하였다. 선정된 주제의 개수는 10개이고, 그 주제가 바로 문서들의 범주가 되는 것이다.

다음 단계로 수집된 실험대상 문서들 중 'outlink' 문서들을 수집하였는데 그 방법으로는 각 추출된 문서의 html 구문을 이용하여 〈a HREF〉 태그를 판별, 'outlink'된 문서의 리스트를 작성하였다.

위와 같은 방법으로 작성된 웹 문서 리스트는 모두 Webzip을 이용하여 HTML 태그를 포함한 전문을 수집하였다. 수집한 결과를 바탕으로 '404 오류' 등으로 내용을 포함하고 있지 않은 문서 등을 삭제시켰다.

그 결과 Yahoo의 디렉토리 중 선정된 10개의 주제에서 실험 대상으로 선택된 문서의 수는 〈표 1〉과 같다.

〈표 1〉 실험집단

주 제 범 주		문서 수	outlink 문서 수
Business & Economy	Accounting and Auditing	205	561
	Music	199	480
Health	Nutrients	129	755
	Weight Issues	151	746
Science	Botanical Gardens	161	591
	Star	180	606
Society & Cultural	Naturists & Nudists	164	523
	Wedding Experiences	184	503
Social Science	Cultural Anthropology	203	624
	Ethnic Study	152	762
합 계		1,728	6,151

〈표 1〉의 실험집단은 실제 실험과정에서 학습문서 집합과 실험문서 집합으로 나뉘게 되며, 학습문서 집합을 기준으로 하여 실험문서들을 범주화하게 된다.

3.2 실험 내용 및 평가 방법

태그 유형을 이용한 자질 추출 단계에서는, 파이썬(python) 프로그래밍 언어를 이용하여 실험집단 내에 있는 문서들의 모든 단어들을 추출한 후, html 구문의 태그를 파악하고 특정 태그에 위치한 단어만을 추출하였다. 추출된 자질로는 모든 단어들, html 구문 중 'title', 'anchor' 태그에 포함된 단어와, 'anchor' 태그을 포함하는 구의 모든 단어 등을 이용하였다. 태그 유형별로

모인 단어들은 단일 자질로 이용되거나, 두 개 이상의 태그 유형을 모아서 하나의 자질로서 실험에 이용되었다.

추출된 단어의 가치는 가중치로 표현하는데, 주어진 문서에만 많이 출현하는 단어일수록 높은 중요도를 가지며, 많은 문서에 출현할수록 낮은 중요도를 가지게 되는 원리를 바탕으로 단어빈도(tf)와 역문헌빈도(idf)를 이용하여 tf·idf 형태의 가중치를 계산하는 것이 일반적이다. 자질의 차원을 축소하기 위해서 본 실험에서는 문헌빈도가 1인 저빈도 단어를 제거하였다. 자질 t의 문서빈도 df 와 전체 문서의 수 n이 주어졌을 때, 이 자질의 가중치 $w(t)$의 공식은 다음과 같다.

$$w(t) = (1 + \log_2 tf) + \log_2(\frac{n}{df})$$

웹 문서 범주화 실험에서는 위의 과정에서 추출된 자질의 가중치를 이용하여 나이브 베이즈 분류기와 kNN 분류기를 사용해서 실제적인 범주화 작업을 수행하였다. 대상 자료를 야후의 디렉토리 서비스를 이용하여 수집하였으므로 모든 학습문서들에는 주제가 할당되어 있는 상태에서 기계학습이 수행된다.

kNN 분류기의 경우 먼저 최적의 k 값을 선택하는 것이 중요하다. 최적의 k 값을 선택하기 위해 k가 13, 10, 5일 경우의 성능을 사전에 실험해 보았다. 그 결과 대체적으로 좋은 성능을 보인 것은 k=13이었다.

복합 분류기의 다양성을 실험하기 위해서 학습문서 집합의 크기를 변형하여 범주화를 시행하였다. 일반적으로 범주화 실험에서는 학습문서가 전체 문서의 70%정도 되지만 본 실험에서는 80%, 70%, 55%의 세 경우로 나누어서 범주화 실험을 하였다.

복합 자질을 대상으로 위의 두 가지 분류기를 이용하여, 학습문서 집합의 크기에 따라 실험한 결과 여러 개의 범주 예측치가 발생하였다. 그 범주 예측치를 하나의 주제 범주로 결정하기 위해 배깅의 방법 중 투표 방식과 가중치합 방식, 최대 신뢰도 방식을 이용하였다.

마지막으로, 두 개의 분류기에서 나온 범주 예측치들을 하나의 값으로 융합하는 분류기 통합을 위한 복합 분류기 실험을 수행하였다. 상이한 분류기를 통해 나온 값들을 융합해야 하므로 배깅 방법 중에서 가중치를 이용한 기법은 제외되었다. 그러므로 본 실험에서는 단일 분류기를 통해 나온 결과물을 융합하는 방법으로 투표 방식을 이용한 복합 분류기를 이용하였다.

본 연구에서 사용한 범주화 성능 평가 척도는 재현율, 정확률, F_1 척도이다.

3.3 범주화 실험

3.3.1 태그유형별 자질 식별 및 자질 결합

본 논문에서는 하이퍼텍스트 태그 유형에 따라 자질을 추출하였는데 그 내용은 아래와 같다.

- all words : 학습 및 실험문서의 모든 단어
- seed_title : 실험문서의 title 및 heading

태그에 있는 단어

- seed_anchor : 실험문서 안에서 하이퍼링크를 나타낼 때 쓰이는 anchor 태그에 있는 단어
- seed_phrase : 실험문서 안에서 하이퍼링크를 포함하는 구의 모든 단어들
- link_title : 실험문서에서 연결된(outlink) 문서의 title 및 heading 태그에 있는 단어
- link_anchor : 실험문서에서 연결된(outlink) 문서의 anchor 태그에 있는 단어

all words에 해당하는 단어들은 wepzip을 통해 수집된 문서들의 html구문만을 제거하여 사용하였고, 그 외의 자질들은 파이썬으로 짜여진 알고리즘에 의해서 추출되었다. 자질 추출을 위한 알고리즘은 html 태그를 이용하였는데, seed_title나 link_title는 〈title〉로 시작하여 〈/title〉로 끝나는 텍스트 사이에 있는 모든 단어를 가져오게 하고, 〈a HERF〉와 〈/a〉사이에 있는 단어들은 seed_anchor와 link_anchor에 속하게 된다. 링크를 포함하는 구의 모든 단어들(seed_phrase)을 추출을 위해서는 그 단어들이 가지는 가치를 이해해야 한다. 예를 들면, 하이퍼텍스트 상에 "My advisor is 〈a HREF="http:/…"〉Tom Mitchell〈/A〉" 이 있다면 〈a HERF〉와 〈/a〉사이에 있는 단어 뿐만 아니라 그 전에 있는 구(phrase)도 범주화에 중요한 근거가 될 수 있다.

태그 유형별로 추출된 6가지 자질을 결합한 결과 8개의 자질 집합이 생성하였다.

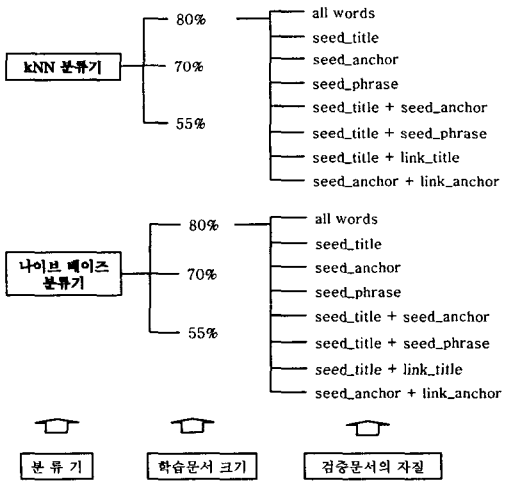
3.3.2 복합 분류기의 범주화 작업

2개의 단일 분류기, 3가지의 학습문서 집단 크기의 변형, 태그 유형별로 추출된 8가지 자질 집합의 결합을 통해 한 문서에 대한 범주 예측치는 다양하게 나올 수 있으므로 여러 개의 예측치를 하나의 결과물로 통합하는 과정을 거쳐야 한다.

〈그림 4〉는 본 실험에서 복합 분류기에 의해 통합되는 48개의 범주화 작업을 정리한 것이다.

4 실험결과 및 분석

〈표 2〉에서 살펴보면 F1 값을 기준으로 가장



〈그림 4〉 복합 분류기의 적용

높은 성능을 보이는 것은 80% 학습문서 집합을 대상으로 kNN 분류기를 이용하여

〈표 2〉 자질 집합에 따른 성능 평가 (F_1 값을 기준으로)

분류기	자질 / 학습집합	a	b	c	d	e	f	g	h
kNN	80%	0.8098	0.7666	0.3636	0.3537	0.7933	0.7852	0.7726	0.6373
	70%	0.7854	0.7775	0.3496	0.3973	0.7717	0.7713	0.7766	0.5865
	55%	0.7705	0.7770	0.3383	0.3367	0.7845	0.7750	0.7733	0.5875
NB	80%	0.5710	0.6522	0.3138	0.3004	0.6171	0.5954	0.6481	0.4669
	70%	0.5278	0.6362	0.3139	0.2852	0.6057	0.5580	0.5954	0.4730
	55%	0.5266	0.6380	0.3114	0.2461	0.6036	0.5660	0.6386	0.4730

* 자질 집합 유형 : all_words(a), seed_title(b), seed_anchor(c), seed_phrase(d), seed_title+seed_anchor(e), seed_title+seed_phrase(f), seed title+link title(g), seed anchor+link anchor(h)

모든 단어를 사용한 경우(a)로서 81.0%의 높은 F_1 값을 보였다.

표제에 출현한 단어(b)는 'anchor' 태그에 속한 단어와 'anchor'를 포함하는 구(phrase)를 독립적으로 사용한 경우(c, d)에 비해 보다 좋은 성능을 나타내었다. c, d의 경우에는 다른 자질과 결합하여 쓰인 경우(e, f, h)에 그 성능이 향상됨을 볼 수 있었다. kNN 분류기를 기준으로 b를 살펴보면, 다른 자질과 결합하여 쓰인 경우(e, f, g)보다는 다소 낮은 성능을 보였지만, 그 값의 차이는 0.3~1.4%로 근소한 것이다. 특히 나이브 베이즈 분류기를 사용하면 결합하지 않고 독립적으로 쓰인 경우(b)가 더 높은 성능을 보이고 있다.

실험문서와 'outlink', 즉 하이퍼링크로 연결된 문서의 자질을 추출하여 자질을 확장시킴으로써 실험 결과가 향상되었다. 실험문서의 자질만 이용한 경우(b, c)보다 하이퍼링크로 연결된 문서의 자질을 통합한 경우(g, h)가 더 좋은 성능을 보였다. 특히 'anchor' 태그에 있는 자질을 기준으로 c와 h의 경우를 비교해 보면, kNN 분류기를 이용한 경우에는 23~27%, 나이브 베이즈 분류기를 이용한 경우에는 15~16% 정도 더 좋은 결과를 나타내었다.

결과적으로, c와 d의 성능은 좋지 않지만 다른 자질과 결합하여 쓰면 그 성능이 향상됨을 알 수 있다. 하지만 e와 f인 경우는 'title' 태그에 위치한 단어들의 긍정적인 영향을 받아서 성능이 향상되는 것이다. 반면, h의 경우는 'anchor'태그에 포함되어 있는 단어만을 사용하여 성능이 향상되었으므로, 이것은 하이퍼링크를 통한 연결문서에 대한 자질 확장이 범주화 성능 향상에 도움이 된다는 것을 증명해 준다.

kNN 분류기에서 가장 높은 F1 값이 81.0%에 비해 나이브 베이즈 분류기에서 가장 높은 F1 값은 65.2%였다. 그러므로 kNN 분류기와 나이브 베이즈 분류기 중 kNN 분류기가 성능이 더 좋다는 것을 알 수 있었다.

자질 집합(a~h)에 대한 실험에서는 한 문서에 대한 범주 예측치가 여러 개 발생한다. 만약 검증 문서에 'anchor' 태그를 가지고 있지 않다면 seed_anchor, seed_phrase, link_anchor 등에 관련된 경우(c, d, h)에서는 범주 예측치가 나올 수 없을 것이다. 그러므로 한 문서에 대한 범주 예측치의 개수는 각각 다르며, 범주 예측치에 따른 가중치의 값이 같은 경우(b=e=f)도 있다.

복합 분류기의 성능은 〈표 3〉에 나와 있다. 자질 집합에 따른 실험과 마찬가지로 나이브 베이즈 분류기보다 kNN 분류기를 바탕으로 한 복합 분류기의 성능이 더 좋은 것으로 나타났다. 이는 자질 집합에 대한 실험의 결과로 나온 여러 개의 범주 예측치를 통합한 것이기 때문에 서로 연관성 있는 결과로 보여진다.

복합 분류기의 범주 예측치 통합 방식에서 kNN 분류기에서는 가중치합 방식이 가장 좋은 성능을 보이고 있으며, 나이브 베이즈 분류기에서는 투표 방식이 가장 좋은 성능을 보이고 있다. kNN 분류기를 이용할 때에는 가중치합 〉 최대신뢰도 〉 투표 방식 순이며, 나이브 베이즈 분류기를 중심으로 보면 투표 〉 최대신뢰도 〉 가중치합 방식 순으로 성능이 나타났다.

kNN 분류기를 사용한 자질 통합 복합 분류기(ensemble classifier based on feature set)에서 가중치합과 최대신뢰도 방식이 좋은 성능을 보이는 것은 선행 연구인 Furnkranz(2002)의 연구와 일치되는 실험 결과이다. 반면, 나이브 베이즈 분류기에서는 투표 방식이 가장 높은 성능을 보이고 있는데, 이는 오분류된 문서들이 많을수록 가중치를 이용하는 방식이 더 불리해지기 때문인 것으로 보인다.

〈표 3〉 자질 결합에 따른 복합 분류기의 성능 평가

(F1 값을 기준으로)

단일 분류기	실험 / 학습집합	최대신뢰도 방식	가중치합 방식	투표 방식
kNN	80%	0.86715	0.87839	0.83546
	70%	0.82724	0.84480	0.81670
	55%	0.83097	0.84059	0.81288
NB	80%	0.64443	0.58494	0.66048
	70%	0.58609	0.54962	0.60663
	55%	0.58012	0.54127	0.60436

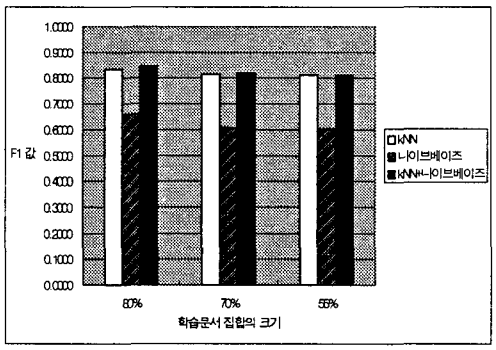
〈표 4〉는 단일 분류기의 결과물들을 융합하기 위한 분류기 통합 복합분류기(ensemble classifier based on single classifier)의 성능을 보여주고 있다.

〈표 4〉 단일 분류기 통합에 따른 복합 분류기의 성능 평가

복합 분류기 / 실험 및 평가		투표 방식 (F_1 값)
kNN 분류기를 기준으로 한 복합 분류기	80%	0.83546
	70%	0.81670
	55%	0.81288
나이브 베이즈 분류기를 기준으로 한 복합 분류기	80%	0.66048
	70%	0.60663
	55%	0.60436
kNN와 나이브 베이즈 분류기를 통합한 복합 분류기	80%	0.84570
	70%	0.81909
	55%	0.81392

〈그림 5〉를 살펴보면, kNN 분류기와 나

이브 베이즈 분류기를 통합한 복합 분류기가 두개의 단일 분류기를 통해 나온 결과에 비해 성능이 높았으며, 이런 결과는 범주 예측치가 많아질수록 복합 분류기의 성능이 향상됨을 보여주고 있다.



〈그림 5〉 kNN과 나이브 베이즈 분류기를 통합한 복합 분류기의 성능

5 결 론

본 연구의 실험 결과는 다음과 같다.

첫째, 하나의 자질만을 독립적으로 이용하는 것보다는 서로 다른 자질들을 결합하여 쓰는 경우 더 좋은 성능을 보인다. 특히 'anchor' 태그에 속한 단어나 'anchor'를 포함하는 구(phrases)를 사용할 때에는 그 자질에 다른 자질을 통합하여 쓴 경우가 더 좋은 성능을 보인다.

둘째, 표제에 출현한 단어만을 자질로 사용했을 때, 매우 우수한 성능을 보이는 것으로 나타났다. 이는 웹 문서의 범주 할당을 위해 문서 내 모든 단어들을 자질로 사용하는 것보다는 html 구문을 이용해 표제를 포함한 몇몇 자질만을 추출하여 그 대상으로 삼는다면 비용과 시간 면에서 더 효율적인 범주화 작업이 될 수 있음을 시사한다.

셋째, 학습문서 집합의 크기와 다양한 자질 집합을 대상으로 한 단일 분류기 실험에서 kNN, 나이브 베이즈 분류기를 사용하였는데, 모든 경우에서 나이브 베이즈 분류기보다 kNN 분류기의 성능이 높은 것으로 나타났다.

넷째, 두 개의 단일 분류기들을 이용하여 8가지의 자질 집합을 대상으로 한 범주화 실험의 결과물들을 통합하기 위해 복합 분류기(ensemble classifier based on feature set)를 사용하였다. kNN 분류기에서는 복합 분류기를 사용하는 것이 단일 자질 집합을 이용할 때보다 언제나 좋은 성능을 나타내었고, 나이브 베이즈 분류기에서는 투표 방식을 이용한 경우에만 더 좋은 성능을 나타내었다.

다섯째, 성능이 좋은 단일 분류기일수록 자질을 통합하는 복합 분류기를 이용하였을 때 더 좋은 성능을 기대할 수 있다. 특히 kNN 분류기를 사용할 경우 복합 분류기 중 가중치를 이용한 통합방식에서 더 좋은 성능을 보였다.

여섯째, kNN 분류기와 나이브 베이즈 분류기를 통합한 복합 분류기(ensemble classifier based on single classifier)가 가장 좋은 성능을 보였으며, 이것은 다중 분류기를 이용하여 나온 다수의 범주 예측치를 통합하는 경우 좀 더 신뢰성 있는 범주가

할당될 수 있기 때문이다.

결론적으로 여러 다른 자질 집합을 사용해 범주화된 결과들을 통합하거나, 2개 이상의 단일 분류기 범주화 결과를 통합하는 복합 분류기를 사용함으로써 범주화 성능을 높일 수 있었다.

참 고 문 헌

이혜원. 2003. **복합 분류기를 이용한 웹 문서 범주화에 관한 연구**. 석사 학위논문, 연세대학교 대학원, 문헌정보학과

Breiman, L. 1996. "Bagging predictors". *Machine Learning*, 26(2): 123-140.

Craven, M., and Slattery, S. 2001. "Relational learning with statistical predicate invention : better models for hypertext". *Machine Learning*, 43(1/2): 97-119.

Dietterich, T. G. 2000. "Ensemble methods in machine learning". *Lecture Notes in Computer Science*. v.1957.

Furnkranz, Johanne. 2002. "Hyperlink ensembles : a case study in hypertext classification". *Information Fusion*, 3: 299-312.

Kuncheva, L. I., Skurichina, M., and Duin, R. P. W. 2002. "An experimental study on diversity for bagging and boosting with linear classifiers". *Information Fusion*, 3: 245-258.

Ledward, A. 1999. "Ensemble Classifiers for machine learning". [cited 2003.6.23]. <http://www-ee.eng.hawaii.edu/~kuh/tmp/ledward.pdf>.

Opitz, D., Maclin, R. 1999. "Popular ensemble methods: An empirical study". *Journal of Artificial Intelligence Research*, 11: 169-198.