점진적으로 계산되는 분류정보와 링크정보를 이용한 하이퍼텍스트 문서 분류 방법

(A Hypertext Categorization Method using Incrementally Computable Class Link Information)

오 효 정 † 맹 성 현 ††
(Hyo-Jung Oh) (Sung-Hyoun Myaeng)

요 약 본 논문은 하이퍼텍스트가 갖는 중요한 특성인 링크 정보를 활용한 문서 분류 모델을 제안한다. 제안된 모델의 주안점은 대상 문서와 링크로 연결된 이웃한 문서의 내용 및 범주를 분석하여 대상 문서 벡터를 조정하고, 이를 근거로 대상 문서가 어느 범주에 해당하는지를 결정한다. 또한, 이웃 문서에 포함된 용어를 반영함으로써 대상 문서의 내용을 확장 해석하고, 이웃 문서의 가용 분류 정보가 있는 경우 이를 참조함으로써 정확도 향상을 기한다. 이러한 접근 방법은 일반 웹 환경에 적용할 수 있는데, 특히 하이퍼텍스트를 주제별로 분류하여 관리하는 검색 엔진의 경우 매일 쏟아져 나오는 새로운 문서와 기존 문서간의 링크를 활용함으로써 전체 시스템의 점진적인 분류에 매우 유용하다. 제안된 모델을 검증하기 위하여 Reuter-21578과 계몽사(ETRI-Kyemong) 자료를 대상으로 실험한 결과 최고 18.5%의 성능 향상을 얻었다.

키워드 : 자동문서분류, 하이퍼텍스트 문서 분류, 하이퍼링크 학습

Abstract As WWW grows at an increasing speed, a classifier targeted at hypertext has become in high demand. While document categorization is quite mature, the issue of utilizing hypertext structure and hyperlinks has been relatively unexplored. In this paper, we propose a practical method for enhancing both the speed and the quality of hypertext categorization using hyperlinks. In comparison against a recently proposed technique that appears to be the only one of the kind, we obtained up to 18.5% of improvement in effectiveness while reducing the processing time dramatically. We attempt to explain through experiments what factors contribute to the improvement.

Key words : Automatic document categorization, Hypertext categorization, Hyperlink learning

1. 서 론

최근 하이퍼텍스트를 대상으로 한 다양한 응용들이 붐을 일고 있다. 특히 하이퍼텍스트가 갖는 특성을 기존 모델에 접목시키려는 시도가 늘어나고 있다. 그 중에서도 하이퍼링크는 문서간의 관계를 나타내는 유용한 정보로서 링크를 통해 연결된 두 문서는 내용적으로 관련이 있어 검색에 도움을 준다는 것은 이미 밝혀진 바 있다[1, 2, 3].

† 비 회 원 : 한국전자통신연구원 휴먼정보처리부 연구원
ohj@etri.re.kr
†† 종신회원 : 충남대학교 컴퓨터학과 교수
shmyaeng@cs.chungnam.ac.kr
논문접수 : 2001년 8월 28일
심사완료 : 2002년 5월 3일

이와 병행하여 하루에도 수만 건씩 쏟아져 나오는 하이퍼텍스트를 주제별로 분류하여 관리하기 위한 분류 모델에 관한 연구도 활발히 진행중이다[4, 5, 6]. 분류 결과는 비단 문서를 관리하는 측면뿐 아니라 검색의 효율을 높인다거나 단어의 의미 중의성(ambiguity) 해소를 위해서도 사용된다[7]. 특히 검색 결과를 필터링(filtering)하는 데 문서의 분류 정보를 활용함으로써 사용자에게 보다 정확한 정보를 제공하려는 연구도 시도되고 있다[2, 8, 9].

많은 양의 문서를 관리하고 이를 효율적으로 검색하기 위한 문서 분류 모델에 관한 연구는 이미 오래 전부터 계속되어 왔다[10, 11]. 이들 대부분은 일반 문서(plain document)를 대상으로 문서에 출현한 용어 정보만을 사용하여 분류하기 때문에 웹 상에 존재하는 다양한 형태의 하이퍼텍스트 문서 분류에 적용하기 어렵다.

그러므로 하이퍼텍스트가 갖는 특성을 고려하여 이들 사이의 관계를 복합적으로 분석하는 새로운 문서 분류 모델이 필요하다[4, 5, 12].

본 논문에서는 이러한 과거 연구를 바탕으로 새로운 문서 분류 모델을 제안한다. 이 모델의 주안점은 대상 문서와 링크로 연결된 이웃 문서의 내용 및 범주를 분석하여 대상 문서 벡터를 조정하고, 이를 근거로 대상 문서가 어느 범주에 해당하는지를 결정한다. 이웃 문서에 포함된 용어를 반영함으로써 대상 문서의 내용을 확장 해석하고, 이웃 문서의 가용 분류 정보가 있는 경우 이를 참조함으로써 분류 성능(effectiveness) 향상을 기한다.

개선된 방법의 주요한 특징은 다음과 같다.

- 한 문서에 링크로 연결된 문서의 분류 정보를 참조하여 이에 해당하는 범주의 가중치(weight)를 조절한다.
- 링크로 연결된 문서의 내용을 대표하는 용어를 반영하여 대상 문서 벡터를 조정한다.

제안된 링크 기반 알고리즘은 이웃 문서의 분류 정보를 반영하는 데 있어, 범주가 미리 할당되지 않은 경우를 대비함으로써 일반 웹 환경에 적용 가능하도록 하며, 점진적인 분류를 통해 학습 효과를 얻음으로써 분류기 생성을 자동화할 수 있다. 또한 분류 과정시 발생하는 오류가 전체 분류 결과에 미치는 영향을 최소화함으로써 이를 보상하기 위한 노력을 해소하며, 이로 인해 분류 시간을 단축시킴으로써 분류의 효율(efficiency)을 향상시킨다.

현재 웹 상에 존재하는 하이퍼텍스트 문서에는 내용과 상관없는 링크가 존재하고 그 패턴도 다양하다. 또한 출현한 용어 역시 매우 다양하기 때문에 이를 학습과정에 사용하게 되면 분류기 생성시 사용한 학습 데이타에 따라 전체 성능이 좌우된다. 그러므로 본 논문에서는 이러한 웹 환경을 고려하는 여러 요소들을 정의하고 이를 판별함으로써 보다 견고한(robust) 분류 알고리즘을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 문서 분류와 관련된 이전 연구에 대해 살펴보고, 3장에서는 본 논문에서 제안한 링크 기반 모델에 대해 설명하며, 4장에서는 이를 검증하기 위한 실험 및 결과에 대해 기술한다. 마지막으로 5장에서 결론을 맺도록 한다.

2. 관련연구

문서를 체계적으로 관리하고 효율적으로 제공하기 위한 문서 분류 모델은 기존의 일반 문서(plain document)를 대상으로 한 모델[10, 11, 13]과 웹의 발달로 새롭게 등장한 하이퍼텍스트(hypertext)를 대상으로 한 모델[4, 5, 6, 12]로 나눌 수 있다. 일반 문서 분류 모델의 경우에는 문서에 출현한 용어만을 활용하는 반면 하이퍼텍스트 문서 분류 모델은 하이퍼텍스트가 갖는 특징인 구조 정보와 링크정보를 활용한다는 차이점이 있다. 이에 대해 보다 자세히 설명하도록 한다.

많은 양의 문서를 관리하고 이를 효율적으로 검색하기 위한 문서 분류 모델에 관한 연구는 이미 오래 전부터 계속되어 왔다. 그 중 대표적인 모델로는 크게 규칙 기반 모델(Rule-based model)과 연역적 학습 모델(Inductive learning model), 검색을 활용한 모델로 나뉘어 진다. 먼저 규칙 기반 모델은 학습 문서들에서 나타나는 범주간의 구별된 규칙을 전문가가 찾아주거나 학습을 통해 추출된 규칙을 이용하여 문서를 분류하는 모델이다[11, 14]. 이 모델은 높은 정확도를 보장하는 반면, 문서 분류 규칙을 사람이 작성하기 때문에 규칙 생성에 많은 시간이 소요되며, 일단 구축된 시스템에서 분류 규칙을 확장하는 것이 어렵다는 단점이 있다[11]. 연역적 학습 모델로는 학습 문서에서 자질을 추출하여 이를 확률적인 접근방법으로 사용한 베이지언(Bayesian) 모델[6, 9, 10, 12, 15]과 트리 구조로 표현하여 자질의 유무로 분류 정보를 결정하는 결정 트리(Decision Tree) 모델[16], 학습 문서를 통해 생성된 양성 자질(positive feature)과 음성 자질(negative feature)을 벡터 공간으로 표현하고 이들 차이를 극명하게 하는 벡터(hyperplane)인 지원 벡터(support vector)를 찾는 SVM(Support Vector Machine)이 있다[5, 17, 18]. 이러한 확률 기반 모델은 학습 과정에서 얻은 지식뿐만 아니라 실제 분류하는 과정에서 얻어지는 지식도 계속 분류기의 자질로 추가할 수 있기 때문에 시스템에 대해 점진적인 학습이 가능하다는 장점이 있다[18]. 이와는 달리 정보검색 관점에서 분류할 대상문서를 질의로 보고 이와 유사한 문서를 찾는 방법인 최근린법(K-nearest Neighbor)[4, 13]과 적합성 피드백(relevance feedback)을 기초로 이를 분류에 응용한 Roccio 모델이 있다[8, 15].

최근에는 단일 모델만을 적용하여 문서를 분류하기보다는 이들 방법들을 비교하여 그 특성을 알아내거나[13, 15, 16], 각 모델의 장점을 복합하여 사용함으로써 성능 향상을 꾀하는 연구가 계속되고 있다[19]. 뿐만 아니라 현재 웹 상에 급진적으로 증가하고 있는 하이퍼텍스트를 대상으로 한 연구가 시도되고 있는데, 하이퍼텍스트가 갖는 가장 큰 특징은 하이퍼링크를 통해 이웃 문서를 참조(reference) 혹은 항해(navigate)할 수 있다는

것이다[2, 6]. 또한 웹 문서의 구조를 표현할 수 있는 태그(tag) 정보를 통해 문서 작성자의 작성 의도를 분류에 반영할 수 있다[4].

하이퍼링크를 통해 분류하고자 하는 대상 문서의 이웃 문서에 내재되어 있는 정보를 참조하여 분류 성능(effectiveness) 향상을 꾀하는 연구로써 IBM의 Hyper Class가 있다[6]. HyperClass는 분류할 대상 문서와 이웃 문서를 MRF(Markov Random Field)로 정의하고 이를 이완 라벨(Relaxation Labeling) 기법을 통해 문서를 분류한다[20]. 그러나 이웃 문서의 분류 정보의 신뢰도에 따라 전체 분류기의 성능이 좌우되며 계산 시간이 증가되는 단점이 있다.

따라서, 본 논문에서는 이웃 문서의 분류 정보를 모르는 경우, 가용 범주(available category)를 할당하고 이에 대한 신뢰도를 낮춤으로써 분류 상태에 미치는 영향을 최소화하는 방안을 제시하여 반복 분류로 인해 발생하는 계산 복잡도(computation complexity) 및 시간 복잡도(time complexity)의 오버헤드(overhead)를 최소화하였다. 본 논문에서는 분류하려는 대상 문서뿐만 아니라 대상 문서와 링크로 연결된 문서들도 분류해야 하는 경우가 발생하므로 분류 속도가 가장 빠른 베이지언(Bayesian) 모델을 사용하였다.

3. 링크 기반 분류 모델

3.1 링크 기반 분류 시스템

하이퍼링크는 두 문서간의 관계를 나타내는 중요한 정보로, 이를 통해 기존 모델의 성능 향상을 꾀하는 연구가 발표되고 있다[1, 2, 3]. [3]의 경우 링크 정보를 통해 검색의 신뢰도(effectiveness)를 향상시켰는데, 하이퍼링크를 활용하기 위해서는 기본적으로 다음과 같은 가정을 한다[2].

[가정 1] 두 문서가 링크로 연결되었을 경우, 두 문서는 서로 관련 있는 내용을 갖는다.

[가정 2] 링크로 연결된 두 문서의 저자가 서로 다른 사람일 경우 링크 생성자는 링크의 종착 문서가 내용적으로 가치가 있다고 판단한다.

위의 가정은 비단 정보 검색에만 적용되는 것이 아니라 이를 문서 분류에도 적용시킬 수 있다. 즉, [가정 1]에 의해 분류할 문서와 링크로 연결된 이웃 문서의 분류 정보를 활용하면 분류 신뢰도를 높일 수 있을 것이다. 또한 [가정 2]에 의해 분류할 대상 문서의 링크 연결성을 분석한다면 보다 효율적인 분류를 수행할 수 있을 것이다. 본 논문에서는 이러한 가정을 바탕으로 링크 정보를 활용한 분류 모델을 제시하고 그 성능 및 효율을 검증하도록 한다.

본 논문에서 제안하는 링크 기반 분류 모델은 하이퍼텍스트가 갖는 링크를 활용한다는 점에서 기존의 일반 문서 분류 모델과 차이가 있다. 일반 문서 분류 모델의 경우 문서에 출현하는 용어만을 사용하여 분류하는 반면 링크 기반 분류 모델은 하이퍼텍스트내의 링크 정보를 활용하여 대상 문서와 연결된 문서를 참조함으로써 분류의 정확도를 향상시키는 모델이다.

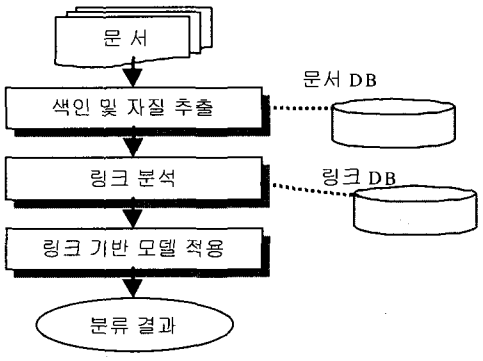

그림 1 링크 기반 분류 시스템의 흐름도

일반적인 문서 분류 시스템은 크게 두 부분으로, 적합한 자질을 추출하여 학습 문서와 분류할 대상 문서 집합, 범주 등을 표현하기 위한 단계인 문서 표현(text representation) 부분과 학습된 분류기를 통해 문서를 어느 범주에 할당할 것인가를 결정하는 문서 분류(text categorization) 부분으로 이루어진다[10]. 또한 문서 분류 과정을 분류기 생성 과정과 분류기 적용 과정으로 나눌 수 있는데, 학습 문서로부터 적합한 자질을 추출하고 그 패턴을 분석하여 분류기를 생성하는 과정을 학습 단계(training phase)라 하고 생성된 분류기를 적용하여 문서를 분류하는 과정을 적용 단계(testing phase)라 한다.

링크 기반 분류 모델은 기존 용어 기반 분류 과정을 따르면서 문서 집합의 링크를 분석하고 활용하는 단계가 더 필요하다. 이는 문서 집합 내에 링크를 분리하여 저장하고 이를 관리하는 디지털 도서관의 환경에서 보다 효율적이다[21]. 그림 1은 링크 기반 분류 과정을 나타낸다. 본 논문은 문서 분류에 관한 연구로써, 분류기의 생성과정에서는 학습 문서가 갖는 링크 정보를 반영하지 않지만 대상 문서를 표현하는 과정에서는 문서가 갖는 용어정보 뿐만 아니라 링크 정보도 함께 표현된다.

3.2 문서 표현(Text Representation)

문서 집합을 색인하고 분류에 필요한 자질을 추출하는

단계로 모든 분류에 공통으로 들어가는 단계이다. 본 논문에서는 문서 집합을 색인하기 위하여 충남대학교 색인기를 사용하였다[22]. 일반적으로 베이지언(Bayesian) 모델의 경우 단어보다는 구(phrase)나 단어 클러스터(cluster)를 사용한 경우의 성능이 높지만[23], 본 연구에서는 단어를 대상으로 자질을 추출하였다.

자질을 추출하기 위해 기대 상호 정보 척도(EMIM: Expected Mutual Information Measure)를 이용하였으며 기대 상호 정보 척도가 임계치(threshold) 이상인 용어만을 추출하여 범주의 중심 벡터(centroid)를 표현한다. 기대 상호 정보 척도를 구하는 식은 다음과 같다[10].

$$I(W_i, C_j) = \sum_{b=0,1}\sum_{c=0,1} P(W_i=b, C_j=c)\log_2\frac{P(W_i=b, C_j=c)}{P(W_i=b)\times P(C_j=c)} \quad (1)$$

일반 문서와는 달리 하이퍼텍스트는 위에서 언급한 용어 정보뿐만 아니라 문서가 가지고 있는 링크 정보, 문서의 구성 형태를 나타내는 구조 정보 등을 이용하여 표현할 수 있다. 본 논문에서 제안하는 링크 기반 분류 모델은 [그림 1]의 링크 분석 과정을 통해 링크 정보를 추출하는데, 추출된 링크는 분류할 대상 문서로 들어오는 링크(in-coming link)와 대상 문서에서 나가는 링크(out-going link)로 나뉜다[1, 3, 21]. 일반 웹 환경에서는 대상 문서에서 나가는 링크인 출력 링크는 추출할 수 있지만, 대상 문서로 들어오는 링크인 입력 링크는 파악하기가 힘들다. 또한 본 논문에서 제안하는 링크 기반 분류 시스템은 링크의 신뢰도를 파악하여 링크 정보의 반영 여부를 결정한다. 즉 링크와 이 링크로 연결된 문서간의 관계를 파악하는 경우가 발생한다. 그러므로 문서 집합내의 링크의 연결성을 분석하고 효율적으로 관리하는 디지털 도서관 환경에서 활용한다면 보다 효과적이다[21].

3.3 베이지언 분류 모델

본 논문에서 사용한 분류 방법은 단순 베이지언(Naive Bayesian) 모델을 이용하였다[6, 9, 10, 12, 15]. 베이지언(Bayesian) 모델은 대상 문서가 각 범주에 속할 확률을 구해 가장 큰 확률 값을 갖는 범주에 그 문서를 할당하는 방법이다. 즉, 문서 d는 $P(c|d)$ 값이 최대가 되는 범주 c에 할당된다. 이를 수식으로 표현하면 다음과 같다[10, 12].

$$Max[P(c|d)] = Max\left[\frac{P(c)P(d|c)}{P(d)}\right] = \underset{c}{Max}\left[P(c)\prod_{i=1}^{n}P(t_i|c)\right] = Max\left[P(c)\prod_{i=1}^{T}P(t_i|c)^{N(t_i,d)}\right] \quad (2)$$

식 (2)에서 $N(t_i|d)$는 문서 d에서 용어 t_i가 출현하는 횟수(tf: term frequency)를 의미하고 T는 전체 문서 집합내의 용어의 수를 나타낸다. 일반적으로 범주 c에 용어 t_i가 많이 나타나고 문서 d에 용어 t_i의 빈도가 높으면 문서 d가 범주 c에 속할 확률이 높다. 그러나 식 (2)를 보면, 용어 t_i가 문서 d에 많이 출현할수록 즉, $N(t_i|d)$가 커질수록 오히려 $P(c|d)$ 값은 작아진다. 이러한 문제를 해결하기 위해 문서가 각 범주에 할당될 확률 값을 구하는 식을 다음과 같이 변형한다[12].

$$P(c)\prod_{i=1}^{T}P(t_i|c)^{N(t_i|d)} \propto \frac{\log P(c)}{n} + \sum_{i=1}^{T}P(t_i|d)\log\left(\frac{P(t_i|c)}{P(t_i|d)}\right) \quad (3)$$

식 (3)의 $P(c)$는 전체 학습문서 집합에서 해당 범주가 나타날 확률을 의미하고 $P(t_i|c)$는 해당 범주에서 용어 t_i가 출현할 확률, $P(t_i|d)$는 대상 문서에서 용어 t_i가 출현할 확률을 의미한다. 문서간의 차이를 나타내기 위해 Kulback-Leiber Divergence 값을 사용하였고, 각각의 범주에 대한 KL Divergence 값을 표현하기 위해 $P(t_i|c)$를 $P(t_i|d)$로 나눠주었다[14]. Kulback-Leiber Divergence는 두 확률 분포의 차이를 교차 엔트로피(cross entropy)를 이용하여 계산하는 방법이다[7].

3.4 링크 기반 분류 모델

3.4.1 링크 기반 분류 모델의 개념

2장에서도 언급했듯이 하이퍼텍스트의 가장 큰 특징 중 하나가 링크를 갖는다는 점이다. 또한 두 문서가 링크로 연결되어 있을 경우, 이 두 문서는 서로 내용적으로 관련이 있다고 가정할 수 있다[1, 2, 3]. 이는 어떤 문서와 링크로 연결된 주위 문서는 같은 범주에 속할 확률이 크다는 것을 의미한다. 그러므로 본 논문에서는 분류할 대상 문서와 이웃한 문서들의 분류 정보를 파악한다면 대상 문서를 보다 정확히 분류할 수 있을 것이라는 가정 하에, 다음과 같은 특징을 갖는 분류 모델을 제시한다.

- **이웃 문서의 분류 정보 반영**
  분류할 대상 문서에 링크로 연결된 문서의 분류 정보를 참조하여 이에 해당하는 범주의 가중치(weight) 조절
- **이웃 문서의 용어 정보 반영**
  대상 문서와 링크로 연결된 문서의 내용을 대표하는 용어를 반영하여 대상 문서 벡터 조정(adjustment)

링크 기반 분류 모델은 분류할 대상 문서의 이웃한 문서에 할당된 범주 정보를 활용하고, 이웃 문서들이 갖고 있는 용어와 대상 문서의 용어를 비교하여 대상 문서의 벡터의 가중치를 조정으로써 대상 문서를 보다 정확히 표현(rich representation)하게 된다. 이를 통해 문서 표현이 정교해짐에 따라 분류 성능(effectiveness)이 높아진다.

기존의 베이지언 분류 모델은 하이퍼텍스트의 특성을 반영하도록 아래와 같이 확장할 수 있다[6].

주어진 조건

- $\Delta=\{\delta_i, =1,2,\ldots,n\}$=문서 δ_i들의 집합
- $G(\Delta)$=문서 집합 Δ과 문서와 문서사이의 링크로 이루어진 그래프
- $T=\{\tau_i\}$=집합 Δ의 벡터 집합, $\tau_i=\{\tau_{ij}|j=1,2,\ldots n_i\}$ =각 문서 용어 벡터
- $c=\{c_i\}$=집합 Δ에 할당된 범주 집합에 의해 문서 분류 모델은 다음과 같이 표현된다.

$$c_i=\arg\max_c[P(C|G,T)]=\arg\max_c\left[\frac{P(G,T|C)P(C)}{p(G,T)}\right] \quad (4)$$

여기서 τ_i는 이웃 문서들의 용어 정보를 반영하여 분류하고자 하는 대상 문서의 벡터를 확장하고 용어의 가중치를 조절한 벡터가 된다. 결국, 분류란 단순 베이지언 확률을 적용하여 대상 문서 δ_i에 범주 c_i가 할당될 확률이 최대가 되는 범주 c를 구하는 것이다. 이를 수식으로 표현한 것이 식 (5)이다.

$$\arg\max_c[P(N_i,\tau_i|c_i)P(c_i)] = \arg\max_c[P(N_i|c_i)P(\tau_i|c_i)P(c_i)] \quad (5)$$

이때 N_i는 문서 δ_i와 연결된 문서 중에서 범주가 미리 할당되어 있는 문서들의 집합을 의미하는 것으로, HyperClass의 경우 $P(N_i|c_i)$을 학습 문서에 나타난 입력 링크(incoming link)와 출력 링크(outgoing link)와 이웃 문서의 분류정보를 분석하여 구했다. 이는 본 논문에서 제시한 모델이 학습 문서내의 용어만을 학습하는 것과는 달리 HyperClass는 학습 문서내의 용어 정보뿐만 아니라 링크 정보도 학습함을 의미한다

본 논문은 기존의 베이지언 모델에 링크 정보를 활용하여 이웃 문서의 분류 정보를 반영할 수 있도록 식 (4)를 통해 식 (2)를 아래와 같이 수정하였다.

$$\begin{aligned}&\arg\max_c[P(C|G,T)]\\&=\arg\max_c[P(C|T)P(C|G)]\\&=\arg\max_c\left[P(c)\prod_{i=1}^{T}P(t_i|c)^{w(t_i,d)}\times Neighbor_d(c)\right]\end{aligned} \quad (6)$$

식 (6)에 나타난 $Neighbor(c)$는 대상 문서 d와 이웃한 문서들의 분류 정보를 반영한 것으로 다음 식을 통해 계산된다.

$$Neighbor_d(c)=\frac{l_d(c)}{L_d}\times w_L \quad (7)$$

이때 L_d는 대상 문서 d와 연결된 링크의 수를, $l_d(c)$는 범주 c에 해당하는 링크의 수를 의미한다. w_L은 링크의 신뢰도를 의미하는 것으로, 가용 범주의 오류가 분류기 전체 성능에 미치는 영향을 최소화한다. 본 연구에서는 링크 신뢰도로 내부실험 결과 0.7을 사용하였다.

3.4.2 링크 기반 분류 알고리즘

이웃 문서의 분류 정보를 반영하는 알고리즘은 대상문서 d와 링크로 연결된 이웃 문서로 이루어진 집합을 생성하고, 이 집합내의 미리 할당된 분류 정보를 통해 $Neighbor(c)$를 구한다(식 6과 7 참조). 그러나 현재 웹 환경은 이웃 문서의 분류 정보가 미리 결정되어 있는 경우도 있고 분류 정보가 없는 경우가 함께 존재한다. 따라서 본 논문에서는 이러한 일반 웹 환경에 적용 가능한 분류 정보 반영 알고리즘을 제시함으로써 보다 견고한(robust) 분류기를 생성하고자 한다. 자세한 과정은 다음과 같다.

[step 1] 대상 문서(d)와 링크로 연결된 문서 집합(A)을 생성한다. 이때, 신뢰할 만한 링크만을 선택하여 확장함으로써 문서 내용과 상관없는 링크를 배제한다. 본 논문에서는 문서간의 유사도(similarity)를 통해 링크의 신뢰도를 측정하였다.

[step 2] 대상 문서(d)와 이웃한 문서내의 용어를 참조하여 공통된 용어의 가중치를 높여줌으로써 대상문서의 내용을 확장 해석한다. 다음은 이를 수식으로 표현한 것이다.

$$w_j=(1-w_j)\cdot w'_j\cdot\partial \quad (8)$$

이때, ∂는 링크의 반영비율을 의미하는 것으로 이웃 문서의 용어의 가중치를 반영하는 정도를 나타낸다.

[step 3] 문서 집합 A에서 범주가 미리 할당되지 않은 문서에 용어 기반 분류(식 2 활용)를 통해 가용범주(available category)를 할당한다.

[step 4] 문서 집합 A의 범주정보에 신뢰도를 할당한다. 문서내의 용어만을 이용해 할당한 가용 분류 정보는 부분 신뢰하고, 미리 할당된 분류 정보나 링크 정보를 활용한 경우의 분류 정보는 완전 신뢰한다.

[step 5] 문서 집합 A의 분류 정보를 반영하는 함수 $Neighbor(c)$ 값을 식 (7)을 통해 구하고, 이를 반영하는 식 (6)을 통해 대상 문서에 적합한 범주를 결정한다.

제안된 링크기반 분류 모델은 기존의 연구 [6]와 다음과 같은 차이점이 있다.

- 제안된 모델은 현재 할당된 분류 정보가 이후의 문서의 범주를 결정할 때 다시 활용되기 때문에 가용 범주 정보의 정확도에 따라 성능이 달라질 수 있다. 이러한 문제점을 해결하기 위해 가용 범주에

대한 신뢰도를 낮게 설정함으로써 가용 범주로 인해 발생하는 오류가 전체 성능에 미치는 영향을 최소화한다. 이는 Hyper Class가 현재 분류 결과로 인해 이전의 분류 결과가 달라지는 경우 반복분류를 통해 보상하는 것과 대조적이다.

- 링크기반 분류 모델에서는 대상 문서가 갖는 링크의 성격에 따라 분류 결과가 달라진다. 그러므로 두 문서간의 내용과 관련 있는 링크만을 반영하기 위해서는 링크의 신뢰도를 판단해야 하는데, 이를 위해 본 논문에서는 두 문서간의 유사도(similarity)가 임계치(threshold) 이상인 링크만을 선택하였다. HyperClass의 경우 문서와 연결된 모든 링크를 활용하며 오류 발생시 relaxa- tion labeling 기법을 통해 이를 보상한다[20].
- 제안된 모델은 HyperClass와 달리 이웃 문서 내의 모든 용어를 반영하지는 않는다. 왜냐하면 모든 용어를 반영하게 되면 대상 문서의 내용 표현과는 무관한 용어가 문서 벡터에 추가됨에 따라 잡음(noise)이 많아지기 때문이다[6]. 즉 대상 문서 내에 존재하는 용어만을 고려하므로 대상 문서 벡터를 확장(expansion)하는 것이 아니라 문서 벡터의 가중치를 조절(adjustment)함으로써 표현의 정확성을 꾀한다.
- 일반적으로 분류 모델은 학습시 학습 문서 집합의 특성을 반영하게 된다. 링크기반 분류 모델 역시 학습 문서 집합의 링크 패턴을 학습하게 되는데, 이는 학습에 사용된 학습 문서의 질에 따라 전체 분류기의 성능이 영향을 받게 된다. 본 논문에서는 학습 문서 집합의 영향력을 줄이기 위해 학습시 링크 정보를 배제하도록 한다.

제안된 링크 기반 분류 모델은 점진적으로 계산되는 분류 정보를 활용하기 때문에 분류되는 문서의 순서에 따라 성능이 달라질 수 있다. 그러므로 실험 문서 집합의 링크 연결성을 분석하여 링크가 많은 문서, 즉 주위 문서에 영향을 많이 주는 문서부터 미리 할당해 나간다면 가용 범주를 할당하는 경우가 줄어들기 때문에 보다 빠르고 정확한 분류가 이루어 질 수 있다. 본 논문에서 제안하는 하이퍼텍스트 문서 분류 모델은 가용 범주에 대한 신뢰도를 달리하고 링크의 연결성을 분석하여 가장 많은 정보를 포함하고 있는 문서부터 분류하는 방법을 통해 현재의 분류 과정이 이전의 분류 과정에 미치는 영향을 최소화하였다.

4. 실험 및 평가

4.1 실험 방법

본 논문에서 제안한 링크 기반 분류의 성능 및 효율을 검증하기 위해 다음 3가지 실험을 실시하였다.

[실험 1] Reuter-21578과 계몽사 집합을 대상으로 한 기존 용어 기반 분류

[실험 2] 계몽사(ETRI-Kyemong) 집합을 대상으로 한 링크 기반 분류

[실험 3] 제안된 모델의 특징에 따른 분류

[실험 1]은 본 논문에서 비교 기준(baseline)으로 삼은 용어 기반 분류 모델의 객관성을 입증하기 위한 실험이고, [실험 2]는 본 논문에서 제안한 링크 기반 분류의 성능을 알아보기 위한 실험이다. [실험 3]은 본 논문과 유사한 연구인 HyperClass[6]와의 차이점을 알아보기 위한 실험으로 각 모델이 갖는 특징에 따라 그 성능을 비교하였다.

표 1은 실험에 사용된 문서 집합의 특성을 분석한 결과이다.

표 1 실험 문서 집합의 특성

특성 \문서 집합	Reuter-21578	계몽사
문서의 수	21,578	23,113
범주가 할당된 문서의 수	11,367	21,525
범주의 수	135	76
용어의 수	25,574	49,578
특징	중복 분류 되어 있음	182,844개의 링크 정보 포함

Reuter-21578 데이타는 이미 기존 연구[10, 13, 15, 18]에서 많이 사용한 실험 데이타 집합으로 총 21,578건의 문서와 135개의 분류 정보를 갖고 있다. 그러나 Reuter-21578 집합은 분류 정보가 할당된 문서 중 70%에 해당하는 문서가 10개의 범주에 해당하며, 10개 이하의 문서가 할당된 범주도 전체 33%로 문서 집합의 분류 정보가 매우 불균형적이다[13]. 그러므로 본 논문에서는 이 중 10,000개의 문서와 87개의 분류 정보를 실험에 사용하였으며, 문서 표현에 사용한 용어의 수는 불용어와 빈도가 1인 용어를 제거한 25,574개이다.

계몽사(ETRI-Kyemong) 데이타는 23,113건의 문서로 구성되어 있으며 대분류 12 범주와 소분류 76 범주로 분류되어 있다. 또한 182,844개의 링크 정보를 포함하고 있어, 본 논문에서 제안하는 문서 분류 모델의 성능을 평가하기 위한 실험 데이타로 적합하다. 23,113개의 문서 중 분류 정보가 할당된 문서는 21,525건이나 이 중 문서의 내용이 없는 경우를 제외한 20,838개의 문서를 실험에 사용하였다. 그러나 Reuter-21578 데이타와 마찬가지로 특정 범주에 집중적으로 많은 문서가 할당되거나 매우 적은 문서가 할당된 경우가 있으며, 범주의 정의가 명확

하지 않은 경우도 있으므로 본 논문에서는 적당한 수의 문서가 할당되어 있고 문서간의 링크 정보가 충분한 범주 28개를 대상으로 실험하였다. 문서 집합내의 용어의 수는 빈도가 1인 용어를 제거한 49,578개이고, 실험 데이타에 대한 색인은 충남대학교 색인기를 사용하였다[22].

실험 결과에 대한 평가 방법으로는 재현도(recall)와 정확도(precision)를 하나의 수로 표현하여 성능을 나타내는 F-score를 사용하였다. 분류 결과를 평가하기 위한 방법으로는 micro 평균과 macro 평균을 사용하였다[9].

4.2 기존 용어 기반 분류 결과

일반적으로 분류기의 성능은 학습에 사용된 학습 문서의 양과 문서를 표현하기 위해 추출한 자질의 수에 따라 다르게 나타난다. [실험 1]은 본 논문에서 비교 기준으로 삼은 용어 기반 분류기의 최적치를 찾는 실험으로써, 궁극적으로 기존 연구의 분류 성능과의 객관적인 비교를 통해 본 논문의 결과에 대한 객관성을 입증하고자 한다. 표 2는 문서 표현을 위해 추출한 자질(feature)의 수에 따른 분류 결과를 나타낸다. 이때 학습 문서 양(T-level)은 80%를 사용하였다.

실험 결과 분석표에 나타난 것처럼 Reuter-21578 데이타의 경우 문서 표현에 사용된 자질의 수가 전체 분류 성능에 미치는 영향이 매우 작다. 마찬가지로 계몽사 집합은 추출된 자질의 수가 많을수록 다양한 분야의 용어를 포함하므로 분류의 정확도가 향상된다. 본논문에서는 추출된 자질의 수가 많을수록 분류에 소요되는 시간이 증가되는 것을 감안하여, 가장 최적인 Reuter-21578 데이타에 대해서는 500개를, 계몽사 데이타에 대해서는 1500개의 자질을 사용하여 문서를 표현하도록 한다.

표 2 추출된 자질의 수에 따른 용어 기반 분류 결과

문서집합	자질의 수	miR	miP	miF	maF
Reuter-21578	500	.8092	.8901	.8495	.6315
	1000	.8021	.8842	.8432	.6029
	1500	.8053	.8982	.8507	.6346
계몽사	1000	.7451			.7131
	1500	.7867			.7618
	2500	.7880			.7484

miR: Micro Avg. Recall, miP: Micro Avg. Precision
miF: Micro Avg. F-score, maF: Macro Avg. F-score

정리해보면, 분류기의 성능은 실험 조건에 따라 달라지기 때문에 이전 연구 성능[10, 22]과 직접적인 비교는 할 수 없지만 Reuter-21578을 대상으로 한 실험 결과 본 논문에서 비교 기준으로 삼은 용어 기반 분류의 성능이 떨어지지 않음을 알 수 있다. 이는 본 논문에서 링크 기반 분류를 통해 얻은 성능 향상이 객관적임을 입증한다. 계몽사 집합에 대한 용어 기반 분류 결과 (*F-score=.7867*)를 [실험 2]의 비교 기준으로 사용하고, 또한 가용 범주에 대한 부분 신뢰도로 이용한다.

4.3 링크 기반 분류 결과

[실험 2]는 본 논문에서 제안하는 링크 기반 분류의 성능을 알아보기 위한 실험으로 [실험 1]의 용어 기반 분류와 비교하였다. 이 실험은 링크 기반 분류에 영향을 주는 요인의 최적치를 찾기 위한 실험으로 분류 성능에 영향을 미치는 다양한 변수를 조합해서 실험하였다. 링크 기반 분류의 성능에 영향을 미치는 요인으로는 다음과 같은 것들이 있으며 이에 따른 실험 결과가 표 3에 나와 있다.

- 링크로 연결된 이웃 문서 용어의 사용 여부
- 링크로 연결된 이웃 문서의 분류 정보 활용 여부
- 링크 신뢰도 정도

[실험 2]는 학습 문서의 양(T-level)으로 80%를, 링크 신뢰도 판단을 위한 유사도 임계치로 0.3을 사용하였다. 결과를 분석해보면 문서내의 용어만을 사용한 경우(*.7867*) 신뢰할 만한 링크를 통해 문서를 확장 해석한 경우(*.7992*)의 성능이 다소 높음을 알 수 있다. 또한 이웃 문서의 분류 정보를 사용한 경우(*.8927*)의 성능이 그렇지 않은 경우(*.7867*)에 비해 월등하다는 것을 보여준다. 일반적으로 이웃한 문서의 용어정보(*.7992*) 혹은 분류 정보(*.8814*) 하나만 사용한 경우보다 둘을 조합해서 사용한 경우(*.8927*)의 성능이 우수함을 알 수 있다

표 3 다양한 요인에 따른 링크 기반 분류 결과

분류 방법			mi. F-score
용어 기반 분류(Baseline)			.7867
링크 기반 분류	이웃 문서의 용어 정보만 활용 (w/δ ≥ 0.3)		.7992(+1.58%)
	이웃 문서의 분류 정보만 활용 (w/δ ≥ 0.3)		.8814(+12.04%)
	이웃 문서의 용어와 분류 정보 활용 (w/δ ≥ 0.3)	Random	.8897(+13.09%)
		최다링크 우선	.8927(+13.47%)
	이웃 문서의 용어와 분류 정보 활용 (w/δ ≥ 0.0)	최다링크 우선	.8326 *(.8927 대비 -6.01%)*

δ=문서간의 유사도, sim(di, dj)

본 논문에서 제안하는 링크 기반 분류 모델은 대상 문서의 분류시 이웃한 문서들의 분류 정보를 반영하기 때문에 이웃 문서가 미리 분류되지 않은 경우에는 용어 기반 분류를 통해 낮은 신뢰도 값의 가용 범주(available category)를 할당한다. 그러므로 링크가 많은 문서부터 링크 기반 분류를 통해 점진적으로 확정 범주를 할당한

다면 이후의 대상 문서의 범주 결정에 미치는 영향이 커지므로 보다 정확한 분류가 이루어진다. 그러나 실험결과 실험 집합을 구성할 때 링크 연결성을 분석하여 링크가 많이 있는 문서부터 분류하는 경우(.8927)가 그렇지 않은 경우(.8897)에 비해 다소 높긴 하지만 그 차이가 매우 적게 나타난 것으로 보아, 분류 순서가 성능에 큰 영향을 미치지 않는다고 판단된다. 이는 제안된 링크 기반 분류 모델이 순서에 따라 할당된 분류 정보가 다른 경우 즉, 용어 기반 분류를 통해 할당된 가용 범주에 오류가 발생한 경우 전체 분류 결과에 미치는 영향을 최소화하는 알고리즘을 통해 이를 보상하기 때문이다.

실험 결과를 종합해보면 신뢰할 만한 링크를 사용하여 이웃 문서의 용어를 통해 대상 문서를 보다 정확히 표현하고, 이웃 문서의 분류 정보를 반영하는 경우가 링크 기반 분류의 최적치임을 알 수 있다.

4.4 제안된 모델의 특징에 따른 분류

본 논문에서 제안하는 링크 기반 모델은 링크의 신뢰도를 판단하여 적합한 링크를 선택한 후 이 링크로 연결된 이웃 문서의 용어 정보를 반영하여 대상 문서 벡터를 조정(adjustment)하고 분류 정보를 반영함으로써 분류 성능의 정확도 향상을 꾀한다. 이와 유사한 연구인 IBM의 HyperClass는 학습 데이타의 링크 정보를 분석하여 분류기를 생성한 후, 이를 활용하여 이웃 문서의 분류 정보를 반영하고 링크로 연결된 이웃 문서의 모든 용어 정보를 반영하여 대상 문서 벡터를 확장(expansion)함으로써 성능 향상을 기한다[6]. [실험 3]은 용어 기반 분류 모델, HyperClass 분류 모델, 제안된 분류 모델의 비교를 통해 각 모델간의 차이점을 알아보기 위한 실험으로 다음과 4가지 요소에 의해 나누어 실험하였다.

- **링크의 신뢰도 판단**

현재 웹 환경에는 문서의 내용과 관련 없이 만들어지는 링크가 존재한다. 그러므로 문서 집합 내의 링크 중 적합한 링크만을 사용하는 것이 성능 향상에 도움이 된다. 특히 링크 정보를 학습하여 분류기를 생성하는 HyperClass의 경우에는 학습 문서 집합의 링크 패턴에 영향을 받기 때문에 적합한 링크의 선택이 매우 중요하다. 본 논문에서는 링크의 신뢰도를 판단하기 위해 문서간의 유사도를 사용하였다. 실험결과, 문서간의 유사도의 임계치를 0.3으로 사용한 경우 89.27%의 정확도를 보인 반면 모든 링크를 사용한 경우에는 83.26%로 낮게 나타났다. 반면 링크의 신뢰도를 판단하기 위해서는 링크의 시작 문서와 종착 문서간의 유사도를 비교해야 하는 오버헤드(overhead)가 있다. 그러나 표 3의 실험 결과, 링크의 신뢰도를 판단하지 않는 경우(*.8326*)가 최적의 경우(*.8927*)보다 6.01% 성능이 떨어지는 것을 볼 때 이는 꼭 필요한 요소이다. 유사도 계산을 위한 오버헤드를 줄이기 위해 처음 실험 문서 집합의 링크를 분석할 때 연결된 문서간의 유사도를 미리 계산하여 링크베이스를 구축한다면 보다 나은 효율을 얻을 수 있을 것이다[21].

- **이웃 문서의 분류 지식**

일반적으로 분류 시스템의 성능은 학습 문서의 양에 따라 달라진다. 특히 링크 기반 분류는 이웃 문서의 분류 정보를 아는 정도에 따라 성능이 달라지므로 이를 변화하여 실험하였다. 실험 변수로 사용한 학습 문서의 양(T-level)은 전체 문서 집합의 20%(4,309개)와 80%(17,225개)를 사용하였고, 이웃 문서의 분류 정보를 아는 정도(K-level)로는 전혀 알지 못하는 경우에서부터 20%만 아는 경우, 50%, 80%, 모두 아는 경우까지 점차 증가시켜 실험하였다. 실험 결과는 표 4와 같다.

표 4 이웃 문서의 분류 정보를 아는 정도에 따른 분류 정확도

T-level(%)	용어 기반	K-level(%)	링크 기반
20	.7308	0	.8062
		20	.8260
		50	.8410
		80	.8580
		100	.8660
80	.7867	0	.8337
		20	.8540
		50	.8738
		80	.8848
		100	.8927

T-level : 학습 문서의 양
K-level: 이웃 문서의 분류 정보를 아는 정도

표 4를 분석하면 용어 기반 분류(*.7867*)보다 링크 기반 분류의 성능(*.8927*)이 모든 경우에 있어 월등함을 알 수 있다. 또한 학습에 사용한 문서의 양이 많을수록 성능이 높고, 링크 기반 분류의 경우 이웃 문서의 분류 정보를 많이 알수록 성능이 향상되었다. 학습 문서의 양이 20%일 때 용어 기반 분류(*.7308*)보다 이웃 문서의 분류 정보를 전혀 모르는 경우에는 10%(*.8062*), 모두 아는 경우에는 18.5%(*.8660*)의 성능 향상을 보였다. 이는 학습 문서의 양이 80%일 때에도 같은 결과를 나타내는데, 이웃 문서의 분류 정보를 모르는 경우에는 6%(*.8337*), 모두 아는 경우에는 13.5%(*.8927*) 정도 용어 기반 분류(*.7867*)보다 정확도가 높게 나타났다. 비록 학습 문서의 양이 적더라도 대상 문서와 이웃한 문서의 분류 정보를

참조한 링크 기반 분류의 경우(*.8062*)가 많은 문서를 학습한 경우의 용어 기반 분류(*.7867*)보다 성능이 더 좋다. 이는 분류기를 만들 때 사람의 힘을 덜 필요로 하고도 좋은 성능의 분류기를 생성할 수 있음을 의미한다.

현재 많은 검색 엔진이 하이퍼텍스트를 주제별로 관리하고 있다[5, 6]. 이러한 현실에서 제안된 링크 기반 분류 모델은 매일 같이 쏟아져 나오는 새로운 문서와 주제별로 관리된 기존 문서간의 링크를 활용함으로써 전체 시스템의 점진적인 분류에 매우 유용하다. 특히 이웃 문서의 분류 정보를 전혀 모르는 경우에도 용어 기반 분류에 비해 성능이 매우 높게 나타나는데, 이는 이웃 문서의 분류 정보가 미리 할당되지 않았으면 가용 범주를 할당함으로써 전체 실험 집합을 점진적으로 분류해 나가기 때문이다. 이를 통해 링크로 연결된 문서의 분류 정보가 미리 정해 있지 않은 일반 웹 환경에서도 좋은 효과가 기대된다.

• 학습 과정시 링크 정보 활용

본 논문에서 제안하는 분류 모델은 기존의 용어 기반 분류 모델의 분류기 생성과정을 그대로 유지하되 분류 과정에서 링크 정보를 활용함으로써 성능 향상을 꾀하는 반면 HyperClass 모델은 분류기를 생성하는 과정과 분류 과정 모두에서 링크 정보를 활용한다. 그러나 분류기 생성시 링크 정보를 반영하는 경우 학습에 사용된 학습 문서의 질에 따라 전체 분류기의 성능이 영향을 받게 된다. 실험 결과, 학습 과정에서 링크 정보를 배제한 경우(*.8927*)가 링크 정보를 반영한 경우(*.8498*)에 비해 월등함을 알 수 있었다. 이는 본 논문에서 사용한 학습 데이타와 실험 데이타는 모두 랜덤(random)하게 추출되었기 때문에 학습 데이타에 포함된 링크의 패턴이 일관적이지 않기 때문이다. 그러므로 보다 견고한 분류기 생성을 위해 학습시 링크 정보를 배제하는 것이 우월하다는 본 논문의 가설이 타당함을 입증한다. 그러나 여기에는 반론이 있을 수 있는데, 학습 데이타 구축시 일관적이고 적합한 패턴의 링크를 갖는 문서들로 학습 문서 집합을 구성한다면 오히려 링크 정보를 학습한 경우가 더 우월하기 때문이다.

• 이웃 문서의 용어 정보 활용

대상 문서를 확장 해석하는 방법에는 크게 두 가지가 있다. 하나는 이웃 문서의 용어 정보를 모두 반영하여 대상 문서 벡터를 확장(expansion)하는 방법이 있고, 다른 하나는 이웃 문서의 용어 중 대상 문서의 용어에 있는 용어만 반영하여 문서 벡터의 가중치를 조절(adjustment)하는 방법이다. 본 논문에서 제안하는 방법은 후자로, 실험 결과 이웃 문서의 모든 용어정보를 반영한 경우(*.6793*)와 가중치를 조절한 경우(*.8997*)의 성능 차가 매우 크게 나타났다. 이는 기존 용어 기반 분류의 성능(*.7897*)보다도 현저히 낮은 정확도를 보이는데, 그 이유는 계몽사 데이타가 백과사전 식으로 구성되어 있기 때문에 문서에 출현한 용어가 매우 다양하며 그 패턴도 일관적이지 않기 때문이다. 이를 통해 모든 용어를 반영하게 되면 대상 문서의 내용 표현과는 무관한 용어가 문서 벡터에 추가됨에 따라 잡음(noise)이 많아지므로 대상 문서 벡터를 유지하면서 가중치만을 조절하는 것이 효과적임을 알 수 있다.

• 반복 분류를 통한 오류 보정

링크 기반 분류 모델은 초기 분류 정보를 통해 점증적으로 전체 문서 집합에 범주를 할당하는 모델이다. 그러므로 초기 오류를 보정하기 위한 대비가 필요하다. 이를 위해 반복분류를 하는 경우와 가용 범주에 대한 신뢰도를 조절하는 방법이 있다. 표 5는 제안된 분류 모델과 이를 가용 범주의 오류를 역수행(backtracking)을 통해 보정하도록 수정한 모델간의 성능 차이를 알아보기 위한 실험 결과이다. 이를 분석해 보면 반복분류를 수행하도록 수정된 모델의 성능과 제안된 모델의 성능 차가 K-level이 낮은 0과 20의 경우에는 다소 낮기는 하지만, 거의 없음을 알 수 있다. 이는 제안된 분류 알고리즘이 가용 범주에 대한 신뢰도를 낮게 설정함으로써 가용 범주로 인해 발생하는 오류가 전체 성능에 미치는 영향을 최소화하였기 때문이다. 그러므로 오류 보정을 위해 반복분류를 하지 않아도 되기 때문에 분류 시간은 단축된다.

표 5 반복분류를 수행한 경우와 그렇지 않은 경우의 성능 비교

K-level	제안된 모델	반복분류 수행 모델
0	.8337	.8569 (-2.71%)
20	.8540	.8748 (-2.38%)
50	.8738	.8784 (-0.52%)
80	.8848	.8860 (-0.14%)
100	.8927	.8986 (-0.66%)

그림 2와 3은 용어 기반 분류 모델, HyperClass 분류 모델, 제안된 분류 모델의 비교를 통해 각 모델간의 차이점을 알아보기 위한 실험 결과이다. 학습 문서의 양(T-level)이 80%일 때 수행한 결과로, 이웃 문서의 분류 정보를 아는 정도(K-level)에 따른 분류 시간과 분류 정확도를 비교한 것이다.

성능을 분석해 보면, [실험 2]의 결과와 같이 용어만 사용하여 문서를 분류한 경우보다 이웃 문서의 분류 정

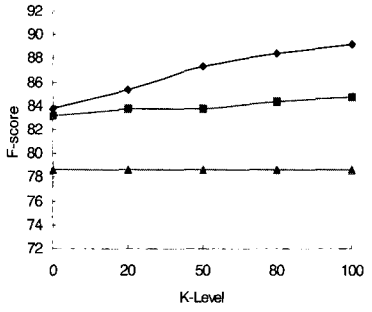

그림 2 세 모델의 분류 성능

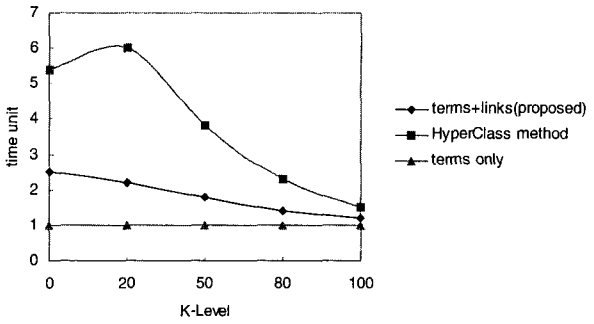

그림 3 세 모델의 분류 시간

표 6 다양한 요인이 분류 정확도에 미치는 영향

특성	HyperClass (Baseline)	제안된 모델	제안된 모델의 성능 향상
학습과정시 링크 정보 활용	○	×	+ 5.0%
링크의 신뢰도 판단	×	○	+ 6.7%
이웃 문서의 분류 정보 활용	○	○	+ 6.6%(제안된 모델)* + 1.9%(HyperClass)*
반복 분류를 통한 오류보정	○	×	- 0.66 ~ - 2.71% (분류 속도/효율성 향상)
이웃 문서의 용어 정보 활용	○	×	+ 31.4%

* Baseline: 이웃 문서의 분류 정보를 전혀 모르는 경우 (즉, K-level = 0)

보를 함께 사용하는 HyperClass 모델과 제안된 모델을 적용한 경우의 성능이 모두 우수함을 볼 수 있다. 또한 HyperClass 모델의 경우 이웃 문서의 범주를 전혀 모르는 경우(*F-score=.8322*)와 모두 아는 경우(*.8480*)의 차가 매우 작은 반면 제안된 모델의 경우는 전혀 모르는 경우(*.8374*)와 모두 알고 있는 경우(*.8927*)의 성능 차가 매우 크다. 그 이유는 HyperClass 모델의 경우 이웃 문서의 범주를 전혀 모르는 경우라 할 지라도 오류 보정을 위해 역수행(backtracking)을 하기 때문에 결국 모든 문서의 범주가 미리 할당된 상태로 수렴하게 되므로 그 성능 차가 크지 않게 된다.

그림 3의 분류 시간을 비교해 보면 용어 기반 분류 모델의 경우 이웃 문서의 분류 정보를 활용하지 않기 때문에 분류 정보를 아는 정도에 따라 분류 시간이 변하지 않는다. 그러나 HyperClass 모델과 제안된 모델의 경우 이웃 문서의 지식의 정도(K-level)에 따라 분류 시간이 많이 달라지는데, HyperClass 모델의 경우에는 이웃 문서의 분류 정보를 모르는 경우 이완 라벨(*relaxation labeling*) 기법을 통해 반복분류하기 때문에 분류 시간이 매우 큰 폭으로 증가하게 된다. 반면 제안된 모델의 경우 이웃 문서의 범주가 미리 할당되지 않은 상태에는 이웃 문서의 분류 정보의 반영비율을 낮추고 미리 할당된 경우에는 반영비율을 높여 점진적으로 분류해 나가기 때문에 반복분류 즉 역수행을 하지 않기 때문에 분류 시간이 크게 증가하지는 않는다.

그림 2를 분석해보면 HyperClass 모델의 성능이 제안된 링크 기반 분류 모델에 비해 전체적으로 성능이 떨어짐을 볼 수 있는데, 이는 사용한 학습 문서 집합내의 링크 정보가 일관적이지 않기 때문이다. 실험에 사용한 계몽사 집합의 링크 정보가 어떤 단어의 정의를 참조하기 위한 목적으로 생성되었기 때문에 링크의 연결 관계가 매우 다양하다. 보다 정교한 학습 문서 집합을 사용하여 분류기를 생성한다면 성능이 매우 향상될 수 있다.

3.4.2에서 언급한 바와 같이 HyperClass 모델과 제안된 모델간의 차이점은 크게 4가지로, 반복 분류 여부와 링크의 신뢰도 판단 여부, 이웃 문서의 용어 활용 정도, 학습 문서 집합 내의 링크 패턴 학습 여부 등이다. 표 6은 이러한 차이가 분류 결과에 어떻게 영향을 미치는지를 보여준다. 예를 들어 학습시 링크 패턴을 학습하지 않는 경우 HyperClass에 비해 제안된 모델이 5%의 성능 향상을 나타냈고, 신뢰할 만한 링크의 선택이 6.7%

의 향상을 나타냈다.

5. 결론 및 향후 연구 방향

본 논문은 하이퍼텍스트의 중요한 특성인 링크를 이용하여 대상 문서를 확장 해석하고 이웃 문서의 분류 정보를 활용하는 링크 기반 분류 모델을 제안하였다. 제안된 모델의 특징은 신뢰할 만한 링크를 통해 이웃 문서의 용어 정보를 선별적으로 선택하여 반영하고, 분류 정보를 참조함으로써 분류 정확도 향상을 기한다. 실험 결과 문서 내의 용어만을 사용하는 기존의 용어 기반 분류 모델에 비해 링크 기반 분류 모델이 최고 18.5%의 성능 향상을 얻을 수 있었다. 또한 가용 범주(available category) 할당을 통해 이웃 문서의 분류 정보를 전혀 모르는 경우를 대비함으로써 일반 웹 환경에 적용 가능성을 보였고, 적은 수의 문서를 학습하고도 이웃 문서의 분류 정보 활용을 통해 높은 성능을 보여줌으로써 분류기 생성을 보다 자동화할 수 있음을 보였다.

폭증하는 하이퍼텍스트 문서를 주어진 시간 내에 적절히 분류하여 검색엔진에 등록시키기 위해서는 효율적인 분류기 생성이 시급하다. 제안된 분류 알고리즘은 이웃 문서의 분류 정보에 대한 신뢰도를 조절함으로써 분류 과정의 오류가 전체 분류 결과에 미치는 영향을 최소화한다. 이를 통해 오류 보정(compensation)을 위한 반복분류 수행 없이도 높은 정확도를 보이는 동시에 67%의 분류 시간 단축 효과를 얻었다. 이러한 알고리즘은 효율적인 분류기 생성에 유리하다.

문서 집합내의 링크 정보가 내용적으로 관련이 있는 링크로 구성되어 있고, 사용된 용어 역시 일관적이라면 학습과정에서 링크 정보를 분석하고 이를 반영하여 분류기를 생성하는 것이 매우 유용할 것이다. 또한 대상 문서의 모든 링크를 통해 이웃 문서에 출현한 모든 용어를 사용하여 대상 문서를 확장 해석함으로써 분류의 정확도 향상을 기할 수 있을 것이다. 그러나 본 논문에서 사용한 계몽사 데이타와 같이 일반 웹 환경에서는 내용과는 상관없는 링크가 존재하며 출현한 용어 역시 매우 다양하다. 제안된 모델은 이를 대비하기 위해 적합한 링크 선택과 이웃 문서 용어에 대한 선별, 학습 과정시 링크 패턴 반영의 배제 등의 요소를 고려함으로써 보다 견고한 분류기 생성에 유리하다.

향후 연구 방향으로는 하이퍼텍스트의 또 다른 중요한 특성인 구조정보를 반영할 수 있도록 제안된 링크 기반 분류 모델을 개선하고 이에 대한 효과를 검증할 예정이다. 특히 실제 웹 문서를 대상으로 제안된 모델을 적용했을 때의 문제점을 파악하고 이를 해결하고자 한다. 또한 링크의 분류 정보를 정보검색에 활용하여 그 효율을 높이는 연구도 수행할 예정이다.

참 고 문 헌

[1] J. M. Lim, H. J. Oh, S. H. Myaeng, and M. H. Lee, "Improving Efficiency with Document Category Information in Link-based Retrieval," *Proc. of the international Workshop on IRAL'99*, 1999.

[2] Kleinberg, J., "Authoritative Source in a Hyperlinked Environment," *Proc. of the 9th annual international ACM-SIAM '98*, 1998

[3] Won-Kyun Joo and Sung-Hyoun Myaeng, "Improving Retrieval Effectiveness with Link Information," *Proc. of the International Workshop on IRAL'98*, 1998.

[4] 정성화, 이종혁, "문서 구조 정보에 기반한 웹 페이지 범주화 모델", *제 10회 한글 및 한국어 정보처리학술대회*, 1998.

[5] Susan Dumais and Hao Chen, "Hierarchical Classification of Web Content," *Proc. of the 23th annual international ACM-SIGIR, July 2000.*

[6] Soumen Chakrabarti, Byron Dom, and Piotr Indyk, "Enhanced Hypertext Categorization using Hyperlinks," *Proc. of the international Conference on SIGMOD '98*, 1998

[7] 이호, 단어 의미 중의성 해결을 위한 분류 정보 모형, *고려대학교 박사학위 논문*, 1999.

[8] Keiichiro Hoashi, Kazunori Matsumoto, Naomi Inoue, and Kazuo Hashimoto, "Document Filtering Method Using Non-Relevant Information Profile," *Proc. of the 23th annual international ACM-SIGIR, July 2001.*

[9] Yu-Hwan Kim, Shang-Yoon Hahn, and Byoung-Tak Zhang, "Text filtering by boosting naive Bayes Classifiers," *Proc. of the 23th annual international ACM-SIGIR, July 2000.*

[10] David D. Lewis, Representation and Learning in Information Retrieval, *Ph.D thesis, Dep. of Computer Science, Univ. of Massachusetts*, 1992.

[11] P. J. Hayes, P. M. Andersen, I. B. Niernburg, and L. M. Schmandt, "TCS: A Shell for Content-Based Text Categorization," *Proc. of the 6th IEEE-CAIA '90*, 1990.

[12] Mark Craven, Dan Di Pasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Sean Slattery, "Learning to Extract Knowledge from the World Wide Web," *Proc. of the international Workshop on AAAI '98*, 1998.

[13] Yiming Yang and Xin Liu, "A Re-examination Of Text Categorization Methods," *Proc. of the 22th annual international ACM-SIGIR*, 1999.

[14] Chidanand Apt , Fred Damerau, and Sholom M. Weis, "Towards Language Independent Automated Learning of Text Categorization models," *Proc. of the 17th annual international ACM-SIGIR*, 1994.
[15] R. E. Shapire, Yoram Singhal, and Amit Singhal, "Boosting and Rocchio applied to text filtering," *Proc. of the 21th annual international ACM-SIGIR, 1998.*
[16] David D. Lewis and Marc Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization," *Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994.
[17] Mart A. Hearst, "Support Vector Machines," *IEEE Information Systems, 13(4):18~28*, 1998
[18] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," *Proc. of the 7th international Conference on CIKM '98*, 1998.
[19] Leah S. Larkey and W. Bruce Croft, "Combining Classifiers in Text Categorization," *Proc. of the 19th annual international ACM-SIGIR 96*, 1996
[20] L. Pelkowitz, "A Continuous Relaxation Labeling Algorithm for Markov Random Fields," *IEEE Trans, on Systems, Man and Cybernetics, 20(3): 705~715*, 1990.
[21] 조은일, 임정묵, 오효정, 이만호, 맹성현, "CORBA와 JAVA를 사용한 에이전트 기반 디지털 도서관 프로토타입 구현", *한국정보과학회 춘계 학술대회*, 1999.
[22] 장동현, 맹성현, "효율적인 색인어 추출을 위한 복합명사 분석방법", *제 8회 한글 및 한국어 정보처리학술대회*, 1996.
[23] L. Douglas Baker and Andrew K. McCallum, "Distributional Clustering of Words for Text Classification," *Proc. of the 21st annual international ACM-SIGIR*, 1998.

오 효 정
1998년 충남대학교 컴퓨터과학과(학사). 2000년 충남대학교 컴퓨터과학화(석사). 2000년 ~ 현재 한국전자통신연구원 휴먼정보검색연구팀 연구원. 관심분야는 문서자동분류, 정보검색, 자연어처리, 기계학습

맹 성 현
1983년 미국 캘리포니아 주립대학 학사. 1985년 미국 Southern Methodist University(SMU) 석사. 1987년 미국 Southern Methodist University(SMU) 박사. 1987년 ~ 1988년 미국 Temple University 교수. 1988년 ~ 1994년 미국 Syracuse University 교수. 1994년 ~ 현재 충남대학교 정보통신공학부 교수. 관심분야는 정보검색, 자연어처리, 디지털도서관, 자동요약, 자동분류, 지식관리시스템