오류 학습 문서 제거를 통한 문서 범주화 기법의 성능 향상

(A Text Categorization Method Improved by Removing Noisy Training Documents)

한 형 동 † 고 영 중 †† 서 정 연 †††
(Hyoungdong Han) (Youngjoong Ko) (Jungyun Seo)

요 약 문서 범주화에서 이진 분류를 다중 분류에 적용할 때 일반적으로 '한 범주에 적합-다른 모든 범주에서는 부적합(One-Against-All) 판정 방법'을 사용한다. 하지만, 이러한 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'은 한 가지 문제점을 가지는데, 적합(positive) 집합의 문서들은 사람이 직접 범주를 할당한 것이지만 부적합(negative) 집합의 문서들은 사람이 직접 범주를 할당한 것이 아니기 때문에 오류 문서들이 많이 포함될 수 있다는 것이다. 본 논문에서는 이러한 문제점을 해결하기 위해서 슬라이딩 윈도우(sliding window) 기법과 EM 알고리즘을 이진 분류 기반의 문서 범주화에 적용할 것을 제안한다. 제안된 기법은 먼저 슬라이딩 윈도우 기법을 사용하여 오류 문서들을 추출하고 이들을 EM알고리즘을 사용해서 다시 범주를 할당함으로써 이진 분류 기반의 문서 범주화 기법의 성능을 향상시킨다.

키워드 : 문서 범주화, 이진 분류, EM알고리즘, 슬라이딩 윈도우 기법, 범주가 할당되지 않은 문서 집합

Abstract When we apply binary classification to multi-class classification for text categorization, we use the One-Against-All method generally. However, this One-Against-All method has a problem. That is, documents of a negative set are not labeled by human. Thus, they can include many noisy documents in the training data. In this paper, we propose that the Sliding Window technique and the EM algorithm are applied to binary text classification for solving this problem. We here improve binary text classification through extracting noise documents from the training data by the Sliding Window technique and re-assigning categories of these documents using the EM algorithm.

Key words : Text Categorization, Binary Classification, EM Algorithm, Sliding Window Technique, Unlabeled Data

1. 서 론

자동 문서 범주화란 문서의 내용에 기반하여 미리 정의되어 있는 범주(category)에 문서를 자동으로 할당하는 작업이다. 학습 작업을 위해 학습데이타를 구성하는 방법에는 이진 분류 구성(binary setting)과 다중 분류 구성(multi-class setting)이 있다. 이진 분류 구성은 두 개의 범주만을 가지며, 이 두 개의 범주는 '관련성이 있는 것 혹은 적합(relevant or positive)'과 '관련성이 없는 것 혹은 부적합(unrelevant or negative)', 즉, 범주에 속하는 문서와 범주에 속하지 않는 문서들에 대한 구분이다[1]. 그러나 일반적인 분류 작업들은 대체로 2개 이상의 범주를 포함하기 때문에 이러한 다중 분류 구성에 이진 분류를 적용할 때는 한 가지 문제점이 발생한다. 다중 분류에서는 각 범주 별로 적합 문서 집합은 존재하지만 부적합 문서 집합은 존재하지 않는다는 것이다. 따라서 이러한 다중 분류의 문제를 해결하기 위해 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'이 일반적으로 사용되어 왔다.

그림 1은 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'을 사용하여 4가지 범주(정치, 경제, 사회, 그리고 스포츠)를 갖는 다중 분류 구성을 이진 분류 구성으로 변환한 예를 보이고 있다. 그림 1에서 볼 수

· 이 논문은 2004학년도 동아대학교 학술연구비(신진과제)에 의하여 연구되었음
† 정 회 원 : 서강대학교 컴퓨터학과 연구원
derichan@empal.com
†† 정 회 원 : 동아대학교 컴퓨터공학과 교수
yjko@dau.ac.kr
††† 종신회원 : 서강대학교 컴퓨터학과 교수
seojy@sogang.ac.kr
논문접수 : 2005년 3월 10일
심사완료 : 2005년 7월 27일

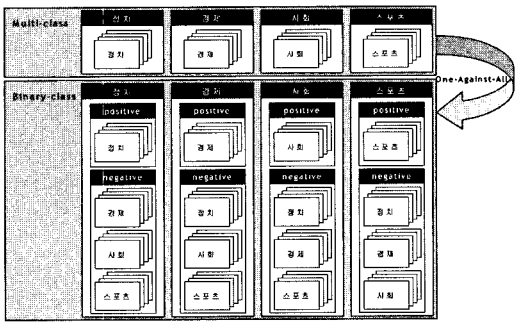

그림 1 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'을 이용한 학습데이타의 재구성

있듯이 적합 집합의 문서들은 각 범주에 대해 사람이 직접 범주를 할당한 것이지만, 부적합 집합의 문서들은 각 범주에 대해 사람이 직접 범주를 할당한 것이 아니다. 그러므로, 부적합 집합은 많은 오류 문서(noise documents)들이 포함될 수 있다. 이러한 오류 문서들은 이진 분류 기반의 문서 범주화(binary text classification)의 성능을 떨어뜨리는 주원인이 된다.

본 논문에서는 이러한 오류 문서들을 효율적으로 제거하여 문서 범주화의 성능을 향상시키기 위한 새로운 방법을 제시한다. 이를 위해서 다음과 같은 두 가지 문제를 해결해야만 한다.

1) "오류 문서들을 포함하는 경계 부분을 어떻게 찾을 것인가?" : 일반적으로 오류 문서들은 적합 문서들과 부적합 문서들이 만나는 경계 부분에 많이 분포한다. 이러한 경계면을 효율적으로 찾기 위해서 본 논문에서는 슬라이딩 윈도우(sliding window) 기법과 혼잡도(entropy) 이론을 이용하여 혼잡도가 가장 높은 부분을 경계면으로 설정하고 그 사이에 있는 문서들을 모두 오류 문서 후보로 고려한다.

2) "찾은 오류 문서 후보들을 어떻게 처리할 것인가?" : 앞에서 찾은 오류 문서 후보들을 처리하기 위해서 본 논문에서는 EM (Expectation Maximization) 알고리즘을 이용하여 오류 문서 후보들을 적합, 부적합 범주에 다시 할당함으로써 학습 문서들을 다시 제 정렬한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 간단히 소개한다. 제3장에서는 제안된 방법의 각 단계에 대해 자세히 설명하고, 제4장에서는 실험을 통해 나온 결과를 비교, 분석한다. 마지막으로 제5장에서는 결론과 향후 연구에 대해서 기술한다.

2. 관련 연구

Yu는 본 논문에서 다루고자 하는 문제를 해결하고자 부적합 집합을 범주가 할당되지 않은 문서 집합으로 간주한 후 이 문서 집합에 포함된 오류 문서들을 제거하고 확실한 부적합 문서들을 식별하는 PEBL이라는 시스템을 제안하였다[2]. PEBL 시스템은 확실한 부적합 문서들을 식별하여 최종적인 분류기를 얻기 위해서 지지벡터 기계(SVM)를 이용하여 분류를 수행한다. 또한 Yu는 PEBL 시스템을 개선하여 논문 [3]에서 MC (Mapping Convergence) 알고리즘과 SVM기술을 결합한 SVMC(Support Vector Machine Mapping Convergence) 방법을 제안하였다.

Liu는 S-EM이라고 하는 새로운 시스템을 제안하였다[4]. 이 시스템은 베이지언 확률 모델과 EM 알고리즘에 기반한다. 이 시스템의 핵심은 범주가 할당되지 않은 문서 집합으로부터 확실한 부적합 문서들을 식별하기 위해 처음으로 스파이(spy) 기법을 사용한다. 스파이 기법은 정확히 범주가 할당된 소량의 문서를 범주가 할당되지 않은 문서 집합에 속하게 한 후 스파이 문서들을 다시 이전에 할당된 범주로 되돌아오게 하고 범주가 할당되지 않은 문서들 중 그 범주에 대한 확실한 부적합 문서들을 식별하는 기술이다. 그리고, 최종 분류기를 생성해 내기 위해서 EM 알고리즘을 수행한다.

Li의 논문은 Rocchio 모델과 SVM 기술을 결합한 시스템을 제안하였다[5]. 이 시스템은 두 단계로 구성되어 있고, 먼저 Rocchio 모델을 사용하여 범주가 할당되지 않은 문서 집합으로부터 1차적인 부적합 문서들을 추출한다. 그 다음 단계에서 오류 문서들을 제거한 확실한 부적합 문서들을 추출하고 최종적인 분류기를 얻어내기 위해 SVM을 반복적으로 수행한다. 그리고 1단계와 2단계 사이에 더 확실한 오류 문서를 제거하기 위해 K-means 군집 알고리즘을 적용한 실험 결과와 적용하지 않은 실험 결과를 비교하였고, 이 비교 실험 결과 K-means 군집 알고리즘을 사용했을 때 더 나은 성능을 보임을 확인할 수 있었다.

앞에서 살펴 본 관련연구들은 성능의 향상에도 불구하고 학습하는 방법들이 여러 개의 분류기들과 문서 군집화 기법 등을 결합하여 복잡한 시스템구조를 가진다. 그러므로, 실제로 시스템을 이해하고 구현하기가 쉽지 않다는 단점이 있다. 하지만 본 논문에서는 단순한 슬라이딩 윈도우기법과 EM알고리즘만을 사용하여 오류문서들을 제거함으로써 모든 문서 분류기에서 높은 성능을 보일 수 있는 단순하지만 효율적인 기법을 제시한다.

3. 제안한 방법

1절의 그림 1의 예에서와 같이 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법' 사용했을 때에는 부적합 집합에 있는 문서들은 직접 각 범주에 대한 부적합 문서로 할당된 것이 아니기 때문에 오류 문서가 많

이 들어 있다는 문제점이 발생하는데, 이들 오류 문서들을 효율적으로 제거함으로써 문서범주화의 성능을 향상시킬 수 있다. 본 논문에서 이들 오류문서들을 효율적으로 제거하기 위해 다음과 같이 4단계로 구성되어진 새로운 기법을 제안한다: (1) '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법', (2) 예견점수(prediction score) 계산, (3) 경계면을 찾기 위한 슬라이딩 윈도우를 사용한 혼잡도 계산, (4) EM 알고리즘을 통한 오류 문서의 재배치

3.1 한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법

2개 이상의 범주를 갖는 다중 분류를 2개의 범주만을 갖는 이진 분류 방법으로 문서를 구성하여 문서 범주화를 수행하고자 할 때 다중 분류의 경우 적합 문서 집합은 존재하지만 부적합 문서 집합은 존재하지 않는 문제가 있다. 이 문제를 해결하기 위해 일반적으로 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'을 사용한다[1].

'한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'은 지정된 범주에 속하는 문서들을 적합 문서들로 간주하고 지정된 범주에 속하지 않는 문서들을 모두 부적합 문서들로 간주하는 것이다[6-8]. 즉, 그림 1과 같이 하나의 범주에 속하는 문서들을 적합 문서들로 간주하고 이 범주를 제외한 나머지 범주들에 속하는 문서들을 부적합으로 간주하는 것이다.

3.2 예견 점수 계산

3.2절과 3.3절의 목적은 많은 오류 문서를 갖는 영역의 경계 부분을 찾는 것이다. 이를 위해 먼저 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'을 사용하여 적합 문서 집합과 부적합 문서 집합을 생성한다. 그 후 생성된 이진 분류 학습 데이타로 베이지언(NB) 분류기를 학습하고 다음 공식을 사용하여 각 문서에 대한 예견 점수를 얻어낸다.

$$\text{Prediction\_Score}(c_i \mid d_j) = \frac{P(Positive \mid d_j)}{P(Positive \mid d_j) + P(Negative \mid d_j)} \quad (1)$$

여기서 c_i는 i번째 범주를 의미하며, d_j는 i번째 범주에 속하는 j번째 문서를 의미한다. $P(Positive|d_j)$는 i번째 범주에서 문서 d_j가 positive일 확률을 의미하며, $P(Negative|d_j)$는 i번째 범주에서 문서 d_j가 부적합일 확률을 의미한다. 이 계산된 예견 점수에 따라 각 범주의 문서들은 점수가 높은 순으로 정렬된다. 식 (1)에서의 확률 $P(Positive|d_j)$와 확률 $P(Negative|d_j)$는 다음의 식 (2)와 같이 일반적인 베이지안 확률 계산식으로써 계산한다[9].

$$P(Positive|d_j) = \frac{P(Positive)P(d_j|Positive)}{P(d_j)} = P(Positive)\prod_{i=1}^{T} P(t_i|Positive)^{N(t_i|d_j)} \quad (2)$$

여기서 $N(t_i|d_j)$는 문서 d_j에서의 용어 t_i가 출현하는 빈도를 의미하고 T는 전체 문서 집합 내의 용어의 수를 나타낸다.

3.3 슬라이딩 윈도우를 사용한 혼잡도 계산

여기서 하나의 경계면을 찾는다는 것은 적합 문서와 부적합 문서가 가장 많이 섞이는 경계 구간을 찾는 것이다. 경계 구간을 찾기 위해, 본 논문에서는 먼저 슬라이딩 윈도우 기법을 사용한다. 일정한 크기를 갖는 윈도우들은 정렬된 예견 점수를 갖는 문서 목록에서 첫 번째 문서에서부터 마지막 문서까지 한 단계, 한 단계 씩 내려간다. 혼잡도 값은 각 윈도우 내의 혼잡 정도(적합 문서와 부적합 문서가 섞이는 정도)를 추정하기 위해 계산된다.

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- \quad (3)$$

각 윈도우(S)의 문서들에 대한 분포의 혼잡도는 식 (3)로서 계산한다[10]. 여기서 S는 적합 문서들과 부적합 문서들이 포함된 집합을 의미한다.

본 논문에서는 가장 높은 혼잡도 값을 갖는 두 개의 윈도우를 뽑는다. 먼저 처음으로 가장 높은 혼잡도 값을 갖는 윈도우를 뽑고 두 번째로 마지막으로 가장 높은 혼잡도 값을 갖는 윈도우를 뽑는다. 그 후 최대경계값(max threshold value)은 처음으로 가장 높은 혼잡도 값을 갖는 윈도우 속에서 가장 높은 예견 점수를 갖는 부적합 문서의 예견 점수를 최대경계값으로 정하고, 최소경계값(min threshold value)은 마지막으로 가장 높은 혼잡도 값을 갖는 윈도우 속에서 가장 낮은 예견 점수를 갖는 적합 문서의 예견 점수를 최소 경계값으로 정한다. 그림 2의 왼쪽 그림은 최대, 최소 경계값을 어떻게 찾는지 보여주고 있다.

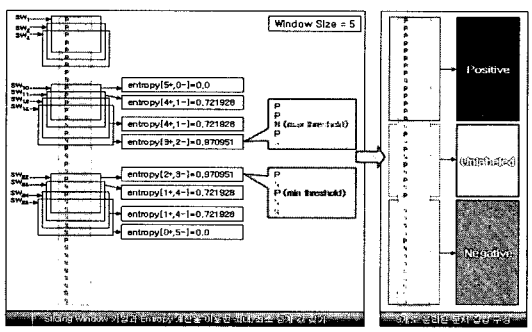

그림 2 경계구간을 찾는 예와 3개의 문서 집합 구성의 예

본 논문에서는 최대, 최소 경계값 사이에 있는 모든 문서들을 오류 문서 후보들로 간주하고 이들을 모두 범주가 할당되지 않는 문서들로 간주한다. 그리고 오류 문서 제거를 위해 이 문서들은 두 개의 범주(적합과 부적합) 중 하나의 범주를 다시 할당 받게 된다.

이로써 그림 2의 오른쪽 그림과 같이 각 범주별 3개의 문서 집합(확실한 적합 문서들, 범주가 할당되지 않은 문서들, 확실한 부적합 문서들)을 갖게 된다. 본 논문에서는 이 3개의 데이타 집합을 EM 알고리즘을 적용하여 각 범주별로 범주가 할당되지 않는 문서들에게 다시 범주를 할당한다.

그러나 이러한 슬라이딩 윈도우 기법의 사용은 다음 3가지 문제에 대해 생각해 봐야 한다.

문제1) "윈도우의 크기는 얼마로 잡아야 하는가?"

문제2) "그림 2의 오른쪽 그림에서 적합 집합에 있는 부적합 문서, 그리고 부적합집합에 있는 적합 문서를 어떻게 처리해야 하는가?"

문제3) "'한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'을 사용했을 때의 적합 문서 집합을 그대로 다 적합 집합으로 간주하는 경우와 그렇지 않은 경우에 둘 중 어느 경우가 더 성능 향상에 더 도움이 되겠는가?"

이들 문제들에 대한 해답은 4.3.1절에서 실험을 통해서 자세히 논의하도록 하겠다.

3.4 EM 알고리즘

본 논문에서 EM 알고리즘은 범주가 할당되지 않는 문서 집합을 잘 정리하고 그 속에 있는 오류 문서들을 제거하기 위해 사용된다. EM 알고리즘은 기대 단계(Expectation step)와 최대화 단계(Maximization step)의 두 단계로 구성되어 있다[11].

EM 알고리즘은 먼저 범주가 할당된 문서(Labeled document)들을 사용하여 분류기를 식 (2)를 이용하여 학습한다. 그 후 범주가 할당되지 않는 문서들에게 범주를 할당한다(Expectation(E or E′) step). 그리고 나서 정리된 학습 데이타를 가지고 다시 분류기를 학습시킨다(Maximization(M) step). 그리고 이 과정((E or E′)-step과 M-step)을 수렴할 때까지 반복하게 된다. 베이지언 분류기에 대해 EM 알고리즘에서 사용되는 단계들은 분류기를 생성하기 위해 사용되는 것과 동일하다. 그림 3은 본 논문에서 EM 알고리즘이 어떻게 사용되는지를 보여준다.

E′-단계는 경계 부근에 있는 오류 문서들을 제거하기 위해 E-단계를 변형한 것이며, 기존의 E-단계와는 달리 적합 집합으로 할당되는 문서들을 오류 문서로 간주하여 제거하게 된다. 결국, 이러한 EM 알고리즘에 의해 새롭게 생성된 이진 학습 데이타를 사용해서 최종적으로 문서 분류기를 학습할 수 있다.

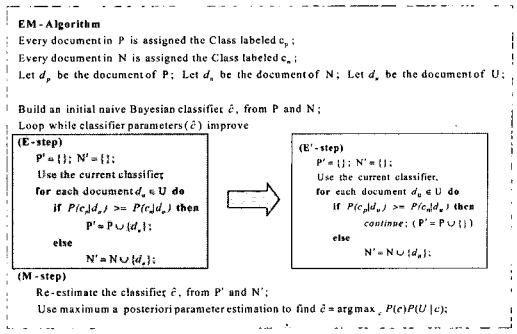

그림 3 EM 알고리즘

4. 실험 및 결과

4.1 실험 데이타 및 실험 환경

실험에서 사용한 테스트 문서 집합은 문서 범주화 영역에서 주로 사용되는 대표적인 두 가지를 사용한다. 첫 번째 문서 집합은 Reuters21578 Distribution 1.0 문서 집합으로써, 12,902개의 기사와 90개의 범주로 구성되어 있으며, 본 논문에서는 다른 연구들에서 가장 많이 사용하는 10개의 범주만을 사용하였다[12]. 그리고 Reuters 문서 집합의 학습 문서 집합 중 약 20%를 검증 문서 집합(validation set)으로 사용하였다. 두 번째 문서 집합은 CMU의 WebKB 프로젝트로부터 생성되었다. 이 문서 집합은 대학의 컴퓨터학과 웹 문서들을 수집한 것으로 웹 문서들은 course, faculty, project, student, department, staff, other의 7개 범주로 나누어져 있다[13]. 본 논문에서는 이 중에 다른 논문들에서 주로 사용하는 4개의 범주들(course, faculty, project, student)을 사용한다[14,15]. 하지만, 뉴스 그룹 문서 집합과 WebKB 문서 집합은 학습 문서와 테스트 문서의 구분이 없으므로, 공정한 평가를 위해서 5중 교차 검증(five-fold cross validation) 기법으로 평가하였다. 즉, 전체의 20%를 테스트 문서로 하고 나머지를 학습문서로 사용하여, 총 다섯 개의 학습 문서와 테스트 문서의 집합을 만들어 각각 실험하고 실험 결과의 평균값으로 성능을 평가하는 기법이다. 불용어 사전을 사용하였으며 스테밍(stemming)은 사용하지 않았다.

실험에서는 제안된 기법의 성능을 비교하기 위하여 일반적으로 많이 사용되는 문서 분류기들을 구현하고 비교하였다. 실험에서 사용된 문서 분류기는 베이지언(Naive Bayes), Rocchio, SVM이다. Rocchio 문서 분류기를 위해서는 α=16 그리고 β=4가 사용되었으며 SVM을 위해서는 SVM^{light} 툴을 이용하여 문서 분류기를 구현하였다.

4.2 성능 평가 방법

평가 방법으로는 정보검색 분야에서 일반적으로 사용되는 이진 분류에 대한 평가 방법인 손익분기점(BEP: BreakEven Point))을 사용하였다. 손익분기점은 정확률(precision)과 재현율(recall)이 같아지는 지점에서의 값을 말한다[16,17]. 모든 범주의 성능을 통합하여 평가하기 위한 기법으로는 문서 범주화 기법의 성능 평가에 주로 사용되는 마이크로 평균(micro-averaging)기법을 사용한다[18].

4.3 실험 결과

이 절에서는 기본 시스템('한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'만 사용한 시스템)과의 비교를 통해 제안된 기법과의 성능을 평가하였다. 제안한 기법을 적용한 시스템의 성능이 모든 분류기를 사용한 시스템에서 그리고 두 개의 문서 집합에서 모두 좋은 성능 향상을 보였다.

4.3.1 4가지 조합의 구성과 EM 알고리즘의 실험

본 실험에서는 경계면을 찾을 때 나타나는 문제1), 문제2), 문제3)에 대한 해답을 얻고자 실험을 통해 평가했으며, 기존 Reuter 데이타의 학습 문서 6,490개를 5:1 비율로 나눠 새로운 학습 문서 5,408개와 검증 문서 1,082개로 구성된 실험 데이타만을 사용하여 실험하였다. 또한 실험에 대한 일관성을 갖기 위해 베이지언 문서 분류기만을 사용하여 실험을 하였다. 여기서, 기본시스템은 기존의 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'을 사용한 것이다.

(1) 4가지 조합에 따른 성능 비교

3.3절의 문제2), 문제3)과 관련하여 표 1의 4가지 조합 구성을 만들어 EM알고리즘의 반복 횟수를 변화해가며 실험하였다. 표 1의 4가지 조합 구성 각각에 3.4절에서 제안한 E′-단계를 적용하여 실험을 하였다.

표 1 4가지 조합 구성

	IOC	***EOC***
OPF	OPF_IOC	OPF_EOC
OPNF	OPNF_IOC	OPNF_EOC

위 표 1에서 ***IOC***(Include Opposite Class)와 ***EOC***(Exclude Opposite Class)는 경계면 부분을 찾을 때 나타나는 문제2)를 해결하기 위한 것이고 ***OPF***(One-Against-All Positive Fix)와 ***OPNF***(One-Against-All Positive Not Fix)는 경계면 부분을 찾을 때 나타나는 문제3)을 해결하기 위한 것이다.

IOC는 반대되는 클래스의 문서를 포함하는 것으로서, 3.2절의 그림 2의 오른쪽 그림에서 적합 집합에 속해 있는 부적합 문서를 적합 문서로 간주하여 적합 집합에 포함시키고 부적합 집합에 속해 있는 적합 문서를 부적합 문서로 간주하여 적합 집합에 포함시키는 것을 의미한다. ***EOC***는 반대되는 클래스의 문서를 포함시키지 않는 것으로서, 그림 2의 오른쪽 그림에서 적합 집합에 있는 부적합문서를 오류 문서로 간주하여 제거하고 부적합 집합에 있는 적합 문서를 오류 문서로 간주하여 제거하는 것을 의미한다.

OPF는 적합 집합은 처음 학습 집합을 구성했을 때 사람이 직접 범주를 할당했기 때문에 EM 알고리즘의 각 반복 단계에서 각 범주의 적합 집합을 그대로 모두 적합 집합으로 간주한다는 의미이고 ***OPNF***는 각 범주의 적합 집합을 그대로 모두 적합 집합으로 간주하지 않고 그림 2와 같이 우리 기법에 의해 생성된 새로운 적합 집합을 사용한다는 의미이다. 즉, ***OPF***를 사용했을 경우에는 EM알고리즘을 사용하기 위해 그림 2와 같이 3가지 문서 집합으로 구분될 때 모든 적합 문서들은 확실한 적합 문서 집합에 속하게 되어 범주가 할당되지 않은 문서 집합에는 어떤 적합 집합도 존재하지 않게 된다.

위 표 1의 4가지 조합 구성 각각에 E′-단계(E′)를 적용해 4가지 조합 구성에 대한 비교 실험을 하였는데, 예를 들어 OPF_EOC의 조합의 경우에는 적합 집합의 문서는 모두 오류문서가 아니라고 보고 적합 집합은 변하지 않고 EM알고리즘에 의해 부적합 문서가 적합문서 집합에 할당되었을 경우에 그들 문서를 오류문서로 인식하여 제거하는 기법이다. 또한, EM 알고리즘에서의 적정한 반복 횟수를 찾기 위해 1부터 10까지 증가시켜가며 실험을 해 보았다. 위 표 1의 실험 모두는 일관성을 갖기 위해 윈도우의 크기를 5로 하여 실험하였으며, 다음 그림 4는 4가지 조합 구성을 실험하여 비교한 성능을 보여주고 있다.

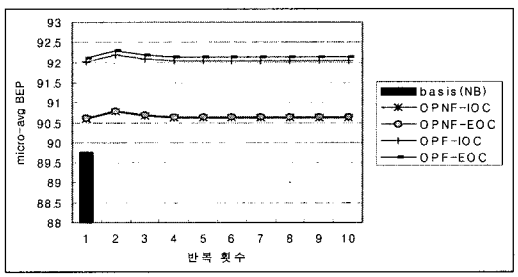

그림 4 4가지 조합과 반복 횟수에 따른 성능 비교

그림 4에서 알 수 있듯이 EOC의 경우가 IOC의 경우보다 더 나은 성능을 보이고 있다. 또한, *OPF*의 경우가 *OPNF*경우보다 훨씬 나은 성능을 보이고 있다. 이는 적합 문서들은 사람이 직접 범주를 할당했기 때문에

오류 문서가 적었기 때문으로 분석된다. 최종적으로 위 그림 4를 통해서 우리는 OPF-EOC의 조합이 가장 좋은 성능을 보인다는 것을 알 수 있었다. 또한 반복 횟수는 2회의 반복 수행을 한 것이 다른 횟수의 반복 수행을 한 것보다 높은 성능을 보인다. 위의 그림 4에서 막대그래프는 검증 문서 집합을 사용했을 때의 기본 시스템의 성능을 평가한 것이며 기본시스템의 성능(89.65%)과 반복 횟수 2회의 조합 OPF-EOC의 성능(92.31%)은 확연한 차이를 보이고 있다.

(2) EM 알고리즘의 E-단계과 E′-단계의 비교

3.4절에서 일반적인 EM알고리즘의 E-단계(E)과 본 논문에서 제안한 E′-단계(E′) 중 어떠한 것이 더 성능 향상에 도움 되는지를 평가하기 위해 다음과 같은 실험을 하였다. 이전 실험에서 가장 성능이 좋았던 조합 OPF-EOC에 E와 E′를 각각 적용하여 비교 실험을 하였고, 실험의 일관성을 위해 윈도우 크기는 5, 반복 횟수 2회로 하여 실험하였다.

표 2의 실험 결과에서 알 수 있듯이 본 논문에서 제안한 E′를 적용한 것이 일반적인 E를 사용한 것보다 전체적으로 우수한 성능을 보임을 확인할 수 있다. 따라서, 이후의 실험부터는 조합 OPF-EOC에 제안한 E′를 적용한 조합 E′-OPF-EOC로서만 실험하고 성능을 평가한다.

(3) 윈도우 크기에 따른 성능 비교

윈도우 크기와 관련된 3.3절의 문제1)에 대한 해답을 얻고자 3, 5, 7의 윈도우 크기를 가지고 비교 실험을 해 보았다. 이 실험은 이전 실험에서 얻은 조합 E′-OPF-EOC와 반복 횟수 2회로 실험하였다.

표 3의 결과에서 알 수 있듯이 윈도우 크기 5일 때가 다른 윈도우 크기 3, 7일 때 보다 92.31로서 더 좋은 성능을 보여주고 있다. 그러므로, 이후의 실험에서는 모두 윈도우 크기 5로 하여 실험하였다.

표 2 EM 알고리즘에서의 E-단계과 E′-단계의 성능 비교

	기본시스템	일반적인 EM (E-단계)	제안한 EM (E′-단계)
micro-avg BEP	89.75	90.52	92.31

표 3 윈도우 크기에 따른 성능 비교

윈도우 크기	3	5	7
micro-avg BEP	91.54	92.31	92.10

4.3.2 분류기 별 성능 비교 실험

본 절에서는 베이지언, Rocchio, 지지 벡터 기계 분류기들을 사용하여 제안된 기법의 성능을 평가하기 위한 실험을 하였다. 표 4, 표 5에서 알 수 있듯이 두 가지 다른 종류의 문서 집합(Reuters, WebKB)에 대해 제안한 시스템이 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'만을 사용한 기본시스템에 비해 모든 범주에서 높은 성능을 보인다. 특히, 베이지언 확률 모델과 Rocchio에서는 뚜렷한 성능 차이를 확인할 수 있다. 또한, 지지 벡터 기계 모델의 경우에도 기본 시스템의 높은 성능을 고려한다면, 성능 향상의 폭을 무시할 수 없을 것이다.

4.3.3 실험 결과 분석 및 토의

본 절에서는 Reuter문서 집합에서의 각 범주의 적합 문서의 수와 제안된 시스템의 성능향상의 정도와의 상관관계를 관찰한다. 표 6에서 보는 바와 같이 일반적으로 적합 문서의 수가 작은 범주에서 성능향상의 정도가 큼을 알 수 있다. 특히 corn, wheat, trade 범주에서 성

표 4 Reuter 데이타에 대한 각 분류기에서의 범주별 성능 비교 표

분류기 / 범주	적합 문서 수	부적합 문서 수	베이지언		Rocchio		SVM	
			기본 시스템	제안 시스템	기본 시스템	제안 시스템	기본 시스템	제안 시스템
acq	1650	4840	96.80	97.89	96.52	97.80	97.64	97.64
corn	181	6309	60.71	71.58	57.14	65.24	87.50	89.29
crude	393	6101	88.89	92.45	87.30	90.15	88.89	91.01
earn	2877	3613	96.96	98.14	96.78	96.87	98.80	98.80
grain	433	6057	89.93	90.94	89.93	91.28	95.30	96.64
interest	347	6143	76.34	76.67	68.70	74.81	84.73	84.73
money	538	5952	79.33	80.90	74.30	77.65	82.68	82.68
ship	197	6293	83.15	92.26	78.65	83.15	88.76	89.89
trade	369	6121	77.12	92.37	60.17	80.73	89.83	90.68
wheat	212	6278	63.38	70.67	66.20	77.83	84.51	85.92
micro-avg BEP			90.80	93.86	89.24	91.80	94.66	95.52
기본시스템 vs. 제안시스템			+3.05		+2.56		+0.86	

표 5 WebKB 데이타에 대한 문서 분류기별 성능 비교

분류기 / 범주	적합 문서 수	부적합 문서 수	베이지언		Rocchio		SVM	
			기본 시스템	제안 시스템	기본 시스템	제안 시스템	기본 시스템	제안 시스템
course BEP	930	3268	83.46	85.67	83.02	86.01	90.15	91.35
faculty BEP	1124	3074	84.25	87.83	84.29	87.90	92.06	92.46
project BEP	503	3695	81.22	83.27	82.69	85.24	89.76	90.43
student BEP	1641	2557	85.39	89.37	84.87	89.00	94.12	94.45
micro-avg BEP			85.67	87.21	86.52	88.26	92.12	92.64
기본시스템vs 제안시스템			+1.54		+1.74		+0.52	

표 6 적합 문서 수에 따른 성능 향상

범주	적합 문서수	부적합 문서수	베이지언	Rocchio	SVM
corn	181	6309	+10.87	+8.1	+1.79
ship	197	6293	+9.11	+4.5	+1.13
wheat	212	6278	+7.29	+11.63	+1.41
interest	347	6143	+0.33	+6.11	0
trade	369	6121	+15.25	+20.56	+0.85
crude	393	6101	+3.56	+2.85	+2.12
grain	433	6057	+1.01	+1.35	+1.34
money	538	5952	+1.57	+3.35	0
acq	1650	4840	+1.09	+1.28	0
earn	2877	3613	+1.18	+0.09	0

능향상의 정도가 매우 컷는데 이는 trade(무역)처럼 범주의 내용이 포괄적이어서 부적합 문서 집합에 오류문서가 많이 발생하는 경우이거나, corn(옥수수)이나 wheat(밀)처럼 서로 비슷한 내용을 담고 있어서 오류문서가 많이 발생하는 경우이다. 결과적으로 제안된 기법은 범주 간 모호성이 심한 범주에서 그리고 적합 학습 문서의 양이 부족한 경우에 더 좋은 성능을 보인다. 실제적으로 문서 범주화를 사용하는 응용 영역에서 좀 더 세분화된 범주화가 필요하거나, 오류 문서가 많거나, 학습 문서의 양을 충분히 확보하지 못하는 경우에 이런 현상은 일반적으로 많이 발생한다. 그러므로, 이러한 경우에 본 논문에서 제안한 기법을 사용한다면 문서 범주화의 성능 향상에 많은 기여를 할 수 있을 것이다.

5. 결론 및 향후 과제

본 논문에서는 이진 문서 분류 시스템에서 주로 사용되는 '한 범주에 적합-다른 모든 범주에서는 부적합 판정 방법'의 문제를 해결하기 위해 슬라이딩 윈도우와 EM 알고리즘을 사용한 새로운 이진 분류 문서 범주화 기법을 제안하였다. 실험을 통해 평가한 결과 제안한 방법이 세 가지 분류기와 두 개의 실험 데이타 모두에서 상당한 성능 향상을 보임을 확인할 수 있었다. 이러한 실험 결과를 바탕으로 제안된 기법을 사용한다면 이진 분류 시스템의 개발에 있어 비교적 간단한 방법을 사용하여 효율적으로 오류 데이타를 제거하고 성능을 올릴 수 있을 것이다.

향후 과제는 다음과 같다. 먼저, 오류 문서 제거를 위해 EM 알고리즘 대신 효과적인 다른 클러스터링 알고리즘 등을 사용할 수 있는지 연구를 진행할 것이다. 또한, 좀 더 정확한 경계면을 찾는 기법을 고려할 것이다.

참 고 문 헌

[1] T. Joachims, *Learning to Classify Text Using Support Vector Machines : theory and Algorithms by Thorsten Joachims*. Dept. of Computer Science, Cornell University, NY, USA, Kluwer Academic Publishers, April, 2002.

[2] H. Yu, J. Han, and K. Chang, "PEBL : Positive Example Based Learning for Web Page Classification Using SVM," *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2002.

[3] H. Yu, C.X. Zhai, and J. Han, "Text Classification from Positive and Unlabeled Documents," *Proceedings of International Conference on Knowledge Management (CIKM'03)*, New Orleans, Louisiana, USA, November 3-8, 2003.

[4] B. Liu, W.S. Lee, P.S. Yu and X. Li., "Partially Supervised Classification of Text Documents," *Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002)*, Sydney, Australia, July 8-12, 2002.

[5] X. Li and B. Liu., "Learning to classify text using positive and unlabeled data," *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, Aug 9-15, 2003.

[6] B. Zadrozny and C. Elkan., "Reducing Multiclass to Binary by Coupling Probability Estimates," *Proceedings of International Conference on Knowledge Discovery and Data Mining(KDD'02)*, 2002.

[7] B. Zadrozny and C. Elkan, "Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers," *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[8] C.-W. Hsu and C.-J. Lin. "A Comparison of Methods for Multi-class Support Vector Machines," *IEEE Transactions on Neural Networks*, 13, pp. 415-425, 2002.

[9] D.D. Lewis, "Naïve (bayes) at Forty: The Independence Assumption in Information Retrieval," *Proceedings of European Conference on Machine Learning*, 1998.

[10] T. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.

[11] A. Demster, N. M. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society series B*, vol 39, No. 1, pp. 1-38, 1997.

[12] K. P. Nigam, *"Using Unlabeled Data to Improve Text Classification,"* Doctoral dissertation, computer Science Department, Carnegie Mellon University, 2001.

[13] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to Construct Knowledge Bases from the World Wide Web," *Artificial Intelligence*, 118 (1-2), pp. 69-113, 2000.

[14] A. McCallum and K. Nigram, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI '98 workshop on Learning for Text Categorization*, 1998.

[15] K. Nigam, A. McCallum, S. Thrun, T. Mitchell, "Learning to Classify Text from Labeled and Unlabeled Documents," *Proceedings of 15th National Conference on Artificial Intelligence (AAAI-98)*, 1998.

[16] Y. Yang, S. Slattery, and R. Ghani. "A Study of Approaches to Hypertext Categorization," *Journal of Intelligent Information Systems*, Vol. 18, No. 2., 2002.

[17] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *ECML*, pp. 137-142, 1998.

[18] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Information Retrieval Journal*, May, 1999.

한 형 동

2002년 건국대학교 컴퓨터과학과 학사 2004년 서강대학교 컴퓨터학과 석사 2004년 11월~현재 (주)코리아와이즈넛 연구소 연구원. 관심분야는 한국어 정보 처리, 자연어 처리, 문서 범주화, 정보 검색, 정보 추출 등

고 영 중

1996년 서강대학교 수학과 학사. 1996년 7월~1997년 12월 LG-EDS 근무. 2000년 서강대학교 컴퓨터학과 석사. 2003년 서강대학교 컴퓨터학과 박사. 2003년 9월~2004년 8월 서강대학교 산업기술연구소 연구원. 2004년 9월~현재 동아대학교 컴퓨터공학과 전임강사. 관심분야는 자연어처리, 텍스트 마이닝, 대화 시스템, 정보 검색, 소프트웨어공학 등

서 정 연

1981년 서강대학교 수학과 학사. 1985년 미국 Univ. of Texas, Austin 전산학과 석사. 1990년 미국 Univ. of Texas, Austin 전산학과 박사. 1990년~1991년 미국 Texas Austin, UniSQL Inc. Senior Researcher. 1991년 한국과학기술원 인공지능 연구 센터 선임연구원. 1991년~1995년 한국과학기술원 전산학과 조교수. 1996년~현재 서강대학교 컴퓨터학과/바이오융합기술 협동과정 정교수. 관심분야는 한국어 정보 처리, 자연어처리, 대화처리, 지능형 정보 검색 등