DOI: 10.3745/KIPSTB.2009.16-B.3.233

확장된 Relief-F 알고리즘을 이용한 소규모 크기 문서의 자동분류

박 흠†

요 약

자질 수가 적은 소규모 크기 문서들의 자동분류는 좋은 성능을 얻기 어렵다. 그 이유는 문서집단 전체의 자질 수는 크지만 단위 문서 내 자질 수가 상대적으로 너무 적기 때문에 문서간 유사도가 너무 낮아 우수한 분류 알고리즘을 적용해도 좋은 성능을 얻지 못한다. 특히 웹 디렉토리 문서들의 자동분류에서나, 디스크 복구 작업에서 유사도 평가와 자동분류로 연결되지 않은 섹터를 연결하는 작업에서와 같은 소규모 크기 문서의 자동분류에서는 좋은 성능을 얻지 못한다. 따라서 본 논문에서는 소규모 크기 문서의 자동분류에서의 문제점을 해결하기 위해 분류 사전작업으로, 예제기반 자질 필터링 방법 Relief-F알고리즘을 소규모 문서 내 자질 필터링에 적합한 ERelief-F 알고리즘을 제시한다. 또 비교 실험을 위해, 기존의 자질 필터링 방법 중 Odds Ratio와 정보이득, 또 Relief-F 알고리즘을 함께 실험하여 분류결과를 비교하였다. 그 결과, ERelief-F 알고리즘을 사용했을 때의 결과가 정보이득과 Odds Ratio, Relief-F보다 월등히 우수한 성능을 보였고 부적절한 자질도 많이 줄일 수 있었다.

키워드 : 자질선택, 자질필터링, 예제기반 필터링, 분류

Document Classification of Small Size Documents Using Extended Relief-F Algorithm

Heum Park†

ABSTRACT

This paper presents an approach to the classifications of small size document using the instance-based feature filtering Relief-F algorithm. In the document classifications, we have not always good classification performances of small size document included a few features. Because total number of feature in the document set is large, but feature count of each document is very small relatively, so the similarities between documents are very low when we use general assessment of similarity and classifiers. Specially, in the cases of the classification of web document in the directory service and the classification of the sectors that cannot connect with the original file after recovery hard-disk, we have not good classification performances. Thus, we propose the Extended Relief-F(ERelief-F) algorithm using instance-based feature filtering algorithm Relief-F to solve problems of Relief-F as preprocess of classification. For the performance comparison, we tested information gain, odds ratio and Relief-F for feature filtering and getting those feature values, and used kNN and SVM classifiers. In the experimental results, the Extended Relief-F(ERelief-F) algorithm, compared with the others, performed best for all of the datasets and reduced many irrelevant features from document sets.

Keywords : Feature Selection, Feature Filtering, Instance-Based Filtering, Classification

1. 서 론

자질 수가 적은 소규모 크기의 문서들의 자동분류는 기존의 분류 알고리즘과 작업공정으로는 좋은 성능을 얻기 어렵다. 그 이유는 문서집단 전체의 자질 수는 크지만 단위 문서 내 자질 수가 상대적으로 너무 적기 때문에, 문서간 유사도 평가에서 유사도 역시 낮아 효과적인 분류 결과를 얻지 못하기 때문이다. 예를 들어 손상된 하드 디스크 복구 작업에서 디스크가 복구 전문프로그램이나 전문가에 의해 복구되었다 하더라도 연결되지 않은 섹터나 잃어 버린 파일의 일부분에 대해서는 수작업이나 섹터간 유사도 평가와 분류알고리즘으로 연결해 주어야 한다. 하지만 섹터의 최대 크기가 512바이트밖에 되지 않아 섹터 내 자질 수가 너무 적어 기존의 유사도 평가에 의한 자동분류 방법으로는 효과적인 결과를 얻지 못한다. 또 포털 사이트의 웹 디렉토리 서비스 문서의 경우 로봇에 의해 수집된 문서들을 전문가에

† 정 회 원 : 부산대학교 컴퓨터공학과 연구원
논문접수 : 2009년 1월 30일
수 정 일 : 1차 2009년 3월 25일
심사완료 : 2009년 3월 25일

의해 수작업으로 분류되어 서비스하지만, 실제 수집된 문서의 많은 수가 사이트의 대문 페이지나 이미지 데이터로 구성되었기 때문에 문서 내 자질 수가 적어 유사도 평가에 의한 자동분류로는 좋은 성능을 얻지 못한다.

이처럼 소규모 크기의 문서들의 자동분류에서는 분류 사전작업으로 문서에서 자질 선택 작업과 문서 내 자질에 따라 자질 값을 새로이 적용해 주어야 한다. 같은 단어라도 문서에 따라 단어(자질)의 대표성 정도가 서로 다르기 때문에, 단순한 자질 빈도에 의한 유사도 측정 방법은 소규모 문서들에게는 적절하지 않다. 따라서 문서 자동분류 사전작업으로, 먼저 자질 필터링 알고리즘을 이용해 문서집합을 대표하는 자질과 클래스별 자질 값을 구했고, 이 대표 자질과 자질 값을 다시 모든 문서에 적용하였다. 그리고 자동분류 알고리즘으로 분류하였다. 분류 전에 자질 선택 작업을 하는 이유는 자질 선택 과정에서 선택된 자질들로 문서집합을 잘 대표하도록 하여 유사도 평가에서 성능을 높이고, 자동분류에서 좋은 성능을 얻기 위해서다. 또한 자질 수가 적은 문서의 자질들은 가중치를 높게 주어 대표 자질로 계속 남게 하고, 자질 수가 많은 문서의 자질들은 부적절한 자질을 제거하도록 필터링 하였다. 기존의 텍스트 문서 자질 선택 방법으로 χ^2, 상호정보 (Mutual Information), 정보이득 (Information Gain), Odds Ratio 등이 많이 사용된다. 정보이득 (IG(t,c))과 Odds Ratio (OR(t,c))의 수식은 아래와 같다[2, 9].

$$IG(t,c) \approx \frac{A}{(A+B)}\frac{A\times N}{(A+C)\times(A+B)} + \frac{C}{(C+D)}\frac{A\times N}{(A+C)\times(C+D)}$$

$$OR(t,c_i) = \frac{\sum_{i=1}^{m}\Pr(t\mid c_i)\sum_{i=1}^{m}\Pr(\bar{t}\mid \bar{c}_i)}{\sum_{i=1}^{m}\Pr(\bar{t}\mid c_i)\sum_{i=1}^{m}\Pr(t\mid \bar{c}_i)} \qquad OR(t,c) \approx \frac{A\times D}{B\times C}$$

A : 범주(문서군) c에 속해 있는 문서 중 단어 t를 포함하고 있는 문서 수

B : 범주(문서군) c에 속하지 않은 문서 중 단어 t를 포함하고 있는 문서 수

C : 범주(문서군) c에 속해 있는 문서 중 단어 t를 포함하고 있지 않은 문서 수

D : 범주(문서군) c에 속하지 않은 문서 중 단어 t를 포함하고 있지 않은 문서 수

N : 전체 문서 수

본 논문에서는 소규모 문서 자동분류를 위해 예제 기반 자질 필터링 알고리즘 Relief-F를 소개하고자 한다. Relief-F 알고리즘은 문서 자질 필터링에서는 시간복잡도(n·m·logm: m은 문서 수, n은 전체 자질 수)가 너무 높아 거의 사용하지 않지만 소규모 크기 문서 내 자질 필터링에서는 효과적일 것이라 판단된다. 그 이유로는 필터링 과정에서 모든 예제(문서)에 대해 자질들 간의 자질 값 차를 가중치로 계산하여 대표 자질을 선택하기 때문에, 자질 수가 적은 예제(문서)의 자질 선택에 가장 적합한 알고리즘이기 때문이다. 하지만 Relief-F 알고리즘은 소규모 크기 문서의 자질 필터링을 위해서는 자질 값 계산 방법과 자질 선택 기준에서 몇 가지 문제점이 있다. 따라서 소규모 문서의 자질 필터링의 문제점을 보안한 ERelief-F 알고리즘을 제안하고자 한다.

실험 결과 비교를 위해, 문서집합에서 tf*idf를 적용한 문서 행렬, Odds Ratio와 정보이득, Relief-F 알고리즘으로 선택한 자질과 자질 값을 적용한 문서 행렬, 또 ERelief-F 알고리즘을 적용한 문서행렬을 함께 실험하였다. 유사도 평가는 코사인측정법을, 자동분류는 kNN과 SVM 분류기를 사용하였다. 그리고 2장에서는 관련연구로 기존의 자질 선택 방법과 예제-기반 자질 필터링 알고리즘 Relief-F에 대해 소개하겠고, 3장은 소규모 문서 내 자질 필터링을 위해 새롭게 구성한 Extended Relief-F (ERelief-F) 알고리즘을 제시하겠다. 또 4장에서는 실험 데이터셋과 실험 방법을 소개하고, 5장은 각 알고리즘 별 실험 결과를 비교 평가하겠다. 마지막으로 6장에서는 결론과 향후 연구과제에 대해 논하겠다.

2. 관련 연구

문서 자동분류에서의 대표적인 자질 선택 방법에는 정보이득(Information Gain), 상호정보(Mutual Information), χ^2, 단어 강도(Term Strength), Odds Ratio 등이 있다 [1][2][9]. 문서 분류에서 이상적인 자질 선택 방법은 자질 수가 적은 문서는 자질 값의 가중치를 높여 자질을 계속 유지시키고, 자질 수가 많고 자질 빈도가 높은 문서는 자질 필터링 작업으로 부적절한 자질들을 제거해야 한다 [2]. 자질 제거 방법에는 자질 래퍼 방법과 자질 필터링 방법이 있다. 자질 래퍼 방법(Wrapper method)은 분류 알고리즘과 함께 학습하면서 분류 성능 평가를 반복적으로 처리해 가장 좋은 자질 집합을 생성하는 것이고, 자질 필터링 방법(Filtering method)은 분류 알고리즘과 독립적으로 자질을 평가하여 자질 집합을 생성하는 것으로 일반적으로 가장 많이 사용한다 [6, 10].

자질 필터링 알고리즘에는 불일치 기준 유전적 자질 선택(Genetic Feature Selection with Inconsistency Criterion: GFSIC)과 Selection Construction and Ranking using Attribute Pattern(SCRAP), RELIEF, 의사결정-트리 필터(Decision-Tree Filter), Cross-Entropy Filter, 예제기반 필터링(Instance Based Filtering), Focus 등이 있다. 또 자질 래핑(Wrapping) 알고리즘에는 Sensitivity-based Feature selection with v-fold Cross Validation(SBFCV), Hill Climbing, Forward Selection, Backward Selection, Bi-Directional Search, Greedy Search, Beam Search, 유전적 알고리즘(Genetic Algorithm) 등이 있다[6, 10].

예제-기반 자질 필터링 방법에는 1992년 Kira&Rendell이 제안한 Relief 알고리즘이 가장 대표적이다. 이 알고리즘이 발표된 이후 많은 예제-기반 자질 필터링 방법들이 개발되었다. 1994년 Kononenko는 Relief 알고리즘의 불완전한 자질 문제와 다중클래스 문제를 해결한 Relief-F 알고리즘을 발표하였다. 그리고 가중치 계산 방법을 개선한 Relief-FF 알고리즘을 비롯해 다양한 가중치 계산 방법과 새로운 알고리즘들이 발표되었다. 2002년 Branidharan Raman은 다양한

Relief 알고리즘의 가중치 계산 방법에 대해 비교 실험한 결과를 발표하였다. 또 2003년 Marko Ronik-Sikonja는 Relief-F algorithm을 회귀 개념을 접목한 Regressional Relief-F 알고리즘을, 2006년 Yijun Sun and Jian Li은 반복적인 Relief 알고리즘을 제시하였다[3-7].

2.1 Relief 알고리즘

Relief 알고리즘의 자질 추정 방법은 먼저 학습 데이터셋에서 주어진 임의의 예제 R_i와 같은 클래스에서 가장 유사한 예제 nearest Hit(H)를 찾고, 다른 클래스에서도 가장 유사한 예제 nearest Miss(M)를 찾는다. 그리고 예제 R_i와 nearest Hit(H) 예제 간의 자질 값 차이와 nearest Miss(M) 예제 간의 자질 값 차이를 이용해 자질의 가중치를 계산한다. 가중치 W(A) 계산에서는 예제 R_i의 자질 A의 값이 같은 클래스 예제 Hit(H)의 자질 값과 차이가 나면 자질 A는 두 예제 간에 공통적 의미가 없으므로 가중치 W(A)를 감소시킨다. 또 예제 R_i의 자질 A의 값이 다른 클래스 예제 Miss(M)의 자질 값과 차이가 나면 자질 A는 두 예제 간에 공통적 의미가 없기 때문에 가중치 W(A)를 증가시킨다. 이 과정을 모든 예제 R_i에 대해 반복하여 자질 A의 가중치 W(A)를 계산한다. 그리고 W(A)가 임계치 θ보다 크면 예제의 대표 자질로 추정한다[3, 4, 6, 7].

여기서 자질 값 차 계산방법에서 자질 값이 연속형이면 자질 값 차 diff(A, I_1, I_2) = |value(A, I_1)-value(A, I_2)|/(max(A)-min(A))로 계산하고, 이산형이면 두 예제가 다를 경우 diff(A, I_1, I_2)=1, 같으면 0으로 계산한다[3, 4]. Relief 알고리즘은 아래와 같다.

Relief 알고리즘

입력: 자질 값과 클래스 정보를 가진 학습 데이터셋
출력: 각 자질의 가중치 *W(A)*로 구성된 실험 데이터셋
모든 자질의 가중치 초기화 *W(A)=0.0*
for *i=1* to *m* do begin
  randomly select instance R_i
  find nearest hit H and nearest miss M
  for *A=1* to *all attribute* do
    $W(A)=W(A)-diff(A,R_i,H)/m+diff(A,R_i,M)/m$
  end
end

2.2 Relief-F 알고리즘

1994년 Kononenko는 Relief 알고리즘의 불완전한 자질 문제와 다중 클래스 데이터셋에 대한 문제 해결 방법으로 Relief-F 알고리즘을 발표하였다. Relief-F 알고리즘은 주어진 임의의 예제 R_i와 같은 클래스의 nearest neighbor Hit 예제 H_j를 유사도 순으로 k개 찾고, 또 다른 클래스의 nearest neighbor Miss 예제 $M_j(C)$를 유사도 순으로 k개를 찾아 자질의 가중치 W(A) 계산에 사용하였다. 가중치 W(A) 계산에서 예제 R_i와 k개의 Hits H_j와의 자질 A 값 차 diff(A, R_i, H_j)의 합에 m과 k로 나눈 값을 빼고, k개의 Misses $M_j(C)$와의 자질 A 값 차 diff(A, R, $M_j(C)$)합에 P(C)/(1-P(class(R_i))의 확률을 곱한 다음 m과 k로 나눈 값을 더해서 계산했다[4, 7]. 여기서 m은 예제 수, k는 nearest neighbor 예제 수다. Relief-F 알고리즘은 아래와 같다.

Relief-F 알고리즘

입력: 자질 값과 클래스 정보를 가진 학습 데이터셋
출력: 각 자질의 가중치 *W(A)*로 구성된 실험 데이터셋
모든 자질의 가중치를 초기화 W(A)=0.0
for i=1 to m do begin
  randomly select instance R_i
  find k nearest hit H_j from class(R_i)
  for each class C≠class(R_i) do
    find k nearest miss $M_j(C)$ from class C
  for A=1 to all attribute do

$$W(A)=W(A)-\sum_{j=1}^{k} diff(A,R_i,H_j)/(m\cdot k)+$$

$$\sum_{C\neq class(R_i)}(P(C)/(1-P(class(R_i)))\sum_{j=1}^{k} diff(A,R_i,M_j(C))/(m\cdot k)$$

  end
end

또 불완전한 자질 데이터 문제를, 즉 두 예제의 자질 중 하나의 자질 값만 있는 경우, 해결하기 위해 자질 값 차 diff(A, I_1, I_2) 계산에서 만약 예제 I_1 또는 I_2의 자질 값을 모르면 diff(A, I_1, I_2)= 1-P(value(A,I_2)|class(I_1)로 계산하였고, 두 예제 자질 값을 모두 모를 경우는 diff(A, I_1, I_2) = $1-\sum_{V}^{\#values}(P(V\mid class(I_1))*(P(V\mid class(I_2))$ 로 계산하였다 (V는 자질 값) [4, 6, 7].

3. ERelief-F 알고리즘

소규모 크기 문서 내 자질 필터링을 위해 Relief-F 알고리즘의 문제점을 보완한 Extended Relief-F (ERelief-F) 알고리즘을 소개하겠다. 자질 필터링에서 기존의 대표적인 자질 선택 방법인 정보이득(Information Gain), 상호정보(Mutual Information), χ^2, Odds Ratio 등이나 Relief-F 알고리즘을 사용하여 소규모 크기 문서에서 자질을 필터링 할 때는 몇 가지 문제점이 있다. 첫째, 기존의 자질 선택 방법인 χ^2, 정보이득과 Odds Ratio를 이용했을 때는 소규모 문서의 경우 선택된 자질과 자질 값이 같은 클래스 내에서는 모두 같으므로 자질의 대표성 평가에서 정확성이 부족하다. 둘째, Relief-F 알고리즘을 이용했을 때는 소규모 문서의 경우 문서 내 자질 수는 적고 전체 자질 수가 크기 때문에, 같은 클래스의 유사 문서 간의 유사도보다 다른 클래스 문서 간의 유사도가 훨씬 더 높아 자질의 가중치가 대부분 0보다 작게 나와 대표 자질을 추출하기 어렵다. 셋째, Relief-F 알고리즘은 앞에서 언급했듯이 소규모 문서라도 자질 필터링에서 시간복잡도(n·m·logm)가 너무 높아 처리 속도가 늦다.

따라서 소규모 크기 문서 자질 필터링에 맞는 자질 가중치 계산 방법이 필요하다. 본 논문에서는 가장 대표적인 예제-기반 필터링 알고리즘인 Relief-F 알고리즘을 이용해, 소규모 크기 문서의 자질 필터링에 맞게 문제점을 보완한 Extended Relief-F (ERelief-F) 알고리즘을 제시하고자 한다. ERelief-F 알고리즘의 기본 개념은 Relief-F 알고리즘을 기반으로 자질 값 평가는 이산형으로 하여 소규모 크기 문서에 맞게 자질 값 차 diff() 계산 방법과 가중치 계산 방법을 수정하였다. 먼저 ERelief-F 알고리즘의 자질 값 차 diff() 계산 방법과 조건은 다음과 같다.

1) 자질 A에 대해, 주어진 문서 R_i의 자질 값 value(A, R_i)와 H_j의 자질 값 value(A, H_j)가 모두 0보다 크면, R_i와 H_j의 자질 값 차는 diff_h()=diff_h()-1.
2) 자질 A에 대해, 주어진 문서 R_i의 자질 값 value(A, R_i)가 0이고 H_j의 자질 값 value(A, H_j)가 0보다 크면, R_i와 H_j의 자질 값 차는 diff_h()=diff_h()+1.
3) 자질 A에 대해, 주어진 문서 R_i의 자질 값 value(A, R_i)가 0보다 크고 M_j의 자질 값 value(A, M_j)가 0이면, R_i와 M_j의 자질 값 차는 diff_m()=diff_m()+1.

여기서 H_j는 문서는 주어진 문서 R_i와 같은 클래스 내에서 가장 가까운 k개 문서이고, M_j는 문서는 주어진 문서 R_i와 다른 클래스 내에서 가장 가까운 k개 문서다. 자질 값 차 diff_h()와 diff_m() 계산 방법과 조건을 수정하게 된 배경은 다음과 같다.

1) 주어진 문서 R_i가 소규모 문서인 경우, 불용어 제거 후 단어 수(3~38개)는 적고 문서집단 전체 단어 수가 상대적으로 많기 때문에 같은 클래스 내 문서 간이라도 유사도는 낮다.
2) 주어진 문서 R_i과 같은 클래스의 문서 수는 다른 클래스의 문서에 비해 상대적으로 적으므로, 같은 클래스 내 문서 간의 유사도는 상대적으로 낮다.
3) 따라서 같은 클래스 내 유사한 문서간 자질 값 차 계산에서는 위 자질 값 차 계산 조건 1)과 2)와 같이 더 명확하게 평가해 주어야 한다.
4) 또 주어진 문서의 자질이 다른 클래스 내 유사한 문서에 없으면 자질 값 차 계산을 위 자질 값 차 계산 조건 3)과 같이 명확하게 평가해 준다.
5) 하지만, 자질이 주어진 문서에는 없고 다른 클래스 내 문서에는 있는 경우에는 자질 값 차 계산에서 제외하였다. 그 이유는 주어진 문서와 유사한 다른 클래스 내 문서가 전반적으로 자질을 더 많이 포함하고 있기 때문이다.

따라서 수정된 자질 값 차 계산 방법을 다시 정리해 보면, 주어진 문서(예제) R_i의 모든 자질에 대해서는 같은 클래스이든 다른 클래스이든 자질 값 차를 모두 평가하였고, 또 주어진 문서 R_i에는 있고 다른 클래스에 없는 경우도 평가하였다. 하지만 주어진 문서 R_i에는 없고 다른 클래스에서 유사도가 높은 문서 Miss(M_j(C]에는 있는 자질에 대해서는 자질 값 차 계산에서 제외했다. 이로 인해 자질 값 계산 처리에서 시간복잡도(n·m·logm) 중 자질 값 차 계산에서 최소 25%이상 줄일 수 있었다. 그리고 자질 A의 가중치 계산은 전체 문서 개수 N번에 대해 자질 값 차를 계산하므로 각 자질 값 차에 N을 나누어 W(A) = W(A) - diff_h()/N + diff_m/N 으로 수정하였다 (N은 전체 문서 수).

모든 문서에서 모든 자질에 대해 자질 가중치 W(A)를 계산한 다음, 가중치가 임계치 Θ이상인 자질만 선택하였다. 그리고 이 자질과 자질 값을 문서 행렬에 다시 적용한 다음 분류에 사용하였다. 새롭게 확장한 ERelief-F 알고리즘 (Extended Relief-F)의 알고리즘은 아래와 같다.

ERelief-F 알고리즘

입력: 자질 값과 클래스 정보를 가진 학습 데이터셋
출력: 각 자질의 가중치 *W(A)*로 구성된 실험 데이터셋
모든 자질의 가중치를 초기화 W(A)=0.0
for i=1 to N do begin
  randomly select instance R_i
  find k nearest hit H_j from class(R_i)
  for each class C≠class(R_i) do
    find k nearest miss $M_j(C)$ from class C
  for A=1 to all attribute do

$$W(A) = W(A) - \sum_{j=1}^{k} diff(A, R_i, H_j)/N + \sum_{j=1, C \neq class(R_i)}^{k} diff(A, R_i, M(C)_j)/N$$

end
for *A=1* to *all attributes* do begin
  if(W(A) > Θ), select attributes
end

주어진 문서 R_i와 같은 클래스와 다른 클래스의 가까운 k개 문서를 찾기 위한 유사도 계산은 각 문서 내 자질의 단어빈도*역문헌빈도 (tf*idf) 값을 이용해 코사인 측정치를 사용하였다.

4 데이터셋과 실험 방법

4.1 실험 데이터셋

실험 데이터셋은 웹 문서 데이터셋과 Reuter-21578 데이터셋을 사용하였다. 웹 문서 데이터셋은 국내 포털 사이트 엠파스, 야후, 네이버의 디렉터리 서비스 중 '자연과학' 디렉터리 중에서 문서 수가 일정 크기 이상인 9개 하위 디렉터리를 선택하여 무작위로 문서를 추출하였다. 엠파스, 야후, 네이버의 디렉터리 서비스 내 문서들은 색인 전문가에 의해 수작업으로 분류되었기 때문에 분류 실험에 대한 결과도 그에 따라 평가하였다. 웹 문서 데이터셋 Empas, Yahoo, Naver의 문서 수는 각각 1,036, 964, 1,069이고, 자질 수는 각각 7,969, 5,063, 7,004 개이고, 클래스 수는 9개다.

그리고 Reuter-21578 데이터셋을 이용해, 손상된 디스크 복구 작업에서 복구 후 연결되지 않은 섹터들을 유사도 계산과 자동분류 알고리즘으로 연결 섹터를 추천하기 위한 실험 환경을 만들었다. Reuter-21578 데이터셋 내 문서 중 크

기가 5kbytes 이상인 파일 174개를 추출하였고, 또 이 파일들을 512bytes 크기로 나누어 총 1,639개의 섹터(문서)를 만들었다 (파일당 평균 9.5개의 섹터). 또 174개의 파일은 3개 그룹으로 나누어 각 58개의 파일씩 3개의 실험 데이터셋을 만들었다. 각 데이터셋의 이름은 Reuter1, Reuter2, Reuter3으로 하였다. 따라서 각 데이터셋은 클래스 수가 각 58개(파일 수와 같음), 즉 같은 파일에서 나온 모든 섹터들은 같은 클래스다. 또 데이터셋별 섹터(문서) 수는 각각 541, 551, 547개, 전체 자질 수는 4,274개, 각 섹터의 최대 크기는 512 바이트다. <표 1>은 데이셋별 문서 수, 클래스 수, 자질 수를 보여준다.

〈표 1〉 데이터셋별 문서 수, 클래스 수, 자질 수

	데이터셋	문서 수	클래스 수	자질 수
Reuter-21578 데이터셋	Reuter1	541	58	4,274
	Reuter2	551	58	4,274
	Reuter3	547	58	4,274
웹 문서 데이터셋	Empas	1,036	9	7,969
	Yahoo	964	9	5,063
	Naver	1,069	9	7,004

4.2 실험 방법

실험은 먼저 자질 선택 알고리즘을 이용해 각 데이터셋의 학습문서로부터 대표 자질과 자질 값을 추출하고, 이 자질과 자질 값을 실험문서에 적용한 다음, 분류기를 이용해 분류결과를 비교하였다. 비교 실험을 위해 ERelief-F 알고리즘과 텍스트 자질 선택에 많이 사용하는 정보이득과 Odds Ratio, 또 Relief-F를 실험하였다. 여기서 대표 자질 선택 기준은 클래스별 자질 값이 임계치 Θ보다 큰 자질만 선택하였다. 가중치 W(A)의 임계치 Θ는 0, 0.0001, 0.0002, 0.0003, 0.0005, 0.0008, 0.0010, 0.0020, 0.0030를 사용하였다. 자동분류 실험을 위해 kNN과 SVM 분류기를 사용했는데, kNN 분류기는 직접 개발하였고 SVM 분류기는 Weka's Explorer의 LIBSVM을 사용하였다. kNN 분류 실험은 5-folds cross validation으로, SVM은 10-folds cross validation으로 하였다. 그리고 유사도 계산은 코사인 측정법을, 분류의 평가는 Mirco-F1과 Macro-F1 척도를 사용하였다. F1은 (2*R*P)/(R+P)로 R은 재현률, P는 정확도다.

5 실험 결과

실험은 자질 선택을 하지 않고 자질 빈도를 tf*idf로 변환한 문서행렬과 정보이득(Information Gain), Odds Ratio, Relief-F를 이용해 선택된 자질을 적용한 문서행렬을 분류하였다 (각각 Org, IG, OR, ReF로 표기). 또 본 논문에서 제시한 Extended Relief-F(ERelief-F)를 이용해 자질 선택을 해 적용한 문서 행렬을 분류한 결과(EReF로 표기)를 비교하였다. 또 각 자질 선택 방법에 따라 선택된 자질 수와 자질 축소율도 비교하였다.

<표 2>는 tf*idf 행렬(Org)로 분류한 결과와, 정보이득(IG)과 Odds Ratio(OR), Relief-F(ReF), ERelief-F(EReF) 등을 이용해 자질을 선택한 다음, kNN분류기로 분류한 결과(k=10)를 데이터셋별로 Micor-F1와 Macro-F1을 비교하였다. 표에서 보듯이 정보이득(IG)과 Odds Ratio(OR), ERelief-F(EReF) 자질 선택 방법을 사용했을 때 아주 높은 성능 향상을 보였다. 또 전체 평균을 보면, Org와 ReF에 비해 EReF의 분류 결과가 월등히 향상된 것을 알 수 있다. <표 3>에서는 SVM 분류기를 사용했을 때 분류 결과를 데이터셋별 Micor-F1와 Macro-F1을 비교하였다. 여기서도 EReF의 분류 결과가 월등히 향상된 것을 볼 수 있다. 특히 웹 문서 데이터셋의 경우 SVM으로 분류했을 때 Relief-F(ReF)의 결과는 tf*idf(Org)의 결과보다 더 낮게 나왔지만 이를 보완한 EReliefF(EReF)의 분류 결과는 월등히 향상된 것을 볼 수 있다. 그리고 정보이득(IG)과 Odds Ratio(OR)보다 더 월등히 향상된 것을 볼 수 있다.

〈표 2〉 tf*idf와 정보이득, Odds Ratio, Relief-F, ERelief-F 알고리즘을 이용했을 때, kNN (k=10)으로 분류한 결과 Micro-F1과 Macro-F1 비교

	DataSets	Org	IG	OR	ReF	EReF
Micro-F1	Reuter1	0.702	0.858	0.856	0.722	0.872
	Reuter2	0.757	0.897	0.906	0.777	0.927
	Reuter3	0.729	0.859	0.872	0.729	0.872
	평균	0.730	0.871	0.878	0.743	0.891
	Empas	0.878	0.835	0.855	0.697	0.919
	Yahoo	0.895	0.855	0.865	0.628	0.935
	Naver	0.792	0.753	0.782	0.604	0.856
	평균	0.855	0.814	0.834	0.643	0.903
Macro-F1	Reuter1	0.709	0.863	0.860	0.725	0.877
	Reuter2	0.761	0.900	0.909	0.774	0.932
	Reuter3	0.736	0.861	0.873	0.728	0.880
	평균	0.735	0.875	0.881	0.742	0.896
	Empas	0.848	0.795	0.820	0.645	0.906
	Yahoo	0.852	0.833	0.843	0.564	0.913
	Naver	0.723	0.714	0.743	0.476	0.839
	평균	0.808	0.781	0.802	0.562	0.886

〈표 3〉 tf*idf와 정보이득, Odds Ratio, Relief-F, ERelief-F 알고리즘을 이용했을 때, SVM으로 분류한 결과 Micro-F1과 Macro-F1 비교

	DataSets	Org	IG	OR	ReF	EReF
Micro-F1	Reuter1	0.786	0.785	0.783	0.626	0.84
	Reuter2	0.770	0.792	0.790	0.484	0.831
	Reuter3	0.644	0.738	0.698	0.479	0.737
	평균	0.733	0.772	0.757	0.530	0.803
	Empas	0.705	0.815	0.819	0.766	0.865
	Yahoo	0.759	0.824	0.832	0.755	0.876
	Naver	0.725	0.796	0.806	0.761	0.815
	평균	0.730	0.812	0.819	0.761	0.852
Macro-F1	Reuter1	0.773	0.779	0.775	0.603	0.821
	Reuter2	0.757	0.778	0.780	0.475	0.821
	Reuter3	0.639	0.711	0.681	0.456	0.715
	평균	0.723	0.756	0.745	0.511	0.786
	Empas	0.689	0.801	0.807	0.757	0.853
	Yahoo	0.739	0.811	0.819	0.740	0.862
	Naver	0.701	0.777	0.788	0.747	0.797
	평균	0.710	0.796	0.805	0.748	0.837

<표 4>는 데이터셋 별로 자질 필터링 후 문서 내 남아 있는 자질 발생 수[1]와 자질 축소율(%)이다. 즉 Org는 자질 필터링을 하지 않은 행렬에서 자질이 전체 문서에서 발생한 수이고, IG와 OR, ReF, EReF는 각각 정보이득, Odds Ratio, Relief-F, ERelief-F를 이용해 자질을 필터링 한 후 문서행렬에 남아 있는 자질 발생 수 (즉, 자질 별 문헌빈도의 합)이다. 여기서 자질 필터링 알고리즘으로 제거된 자질의 축소율을 보면, IG와 OR에서는 Reuter-21578 데이터셋의 경우 자질 축소율은 51~53%지만, 웹 데이터셋에서는 46~79%로 큰 폭을 볼 수 있다. 그리고 ReF 역시 데이터셋별 자질 축소율이 큰 차이를 보이고 있다. 반면 EReF의 경우 전반적으로 40~50%대의 자질 축소율을 보이고 있어 소규모 문서 내의 자질 필터링에서는 안정된 결과를 보였다.

〈표 4〉 데이터셋 별 자질 필터링 후 문서 내 남아 있는 자질 발생 수/축소율(%)

DataSets	Org	IG	OR	ReF	EReF
Reuter1	14,981	7,935/53	8,741/58	10,992/73	7,142/47
Reuter2	15,246	8,050/52	9,057/59	10,721/70	6,742/44
Reuter3	14,962	7,680/51	8,484/56	10,909/72	7,124/47
Empas	65,742	39,124/59	52,280/79	16,207/24	35,585/54
Yahoo	42,077	31,964/76	32,301/76	17,246/41	24,499/58
Naver	56,580	26,387/46	45,585/80	11,977/21	28,691/50

6. 결 론

자질 수가 적은 소규모 크기 문서 자동분류를 위해 예제-기반 자질 필터링 방법인 ERelief-F 알고리즘을 제시하였고, 이 알고리즘을 이용해 선택된 자질과 자질 값을 적용하여 분류한 결과 전반적으로 아주 좋은 분류 결과를 보였다. Relief-F 알고리즘은 주로 데이터 필터링 작업에 많이 사용되었는데 문서 분류에 적용했을 때는, 소규모 문서의 경우 문서간 유사도가 상대적으로 낮아 대표 자질을 추출하는데 실패하였고 그 결과 분류 결과가 tf*idf보다 낮게 나왔다. 또 자질 필터링 작업에서 시간복잡도(n·m·logm)가 너무 높아, 문서 수가 많거나 자질 수가 많은 문서집합의 필터링 작업에서는 부적합하였다. 하지만 ERelief-F 알고리즘을 적용했을 때는 실험 결과에서 보듯이 kNN과 SVM 분류 결과 micro-F1과 macro-F1이 다른 자질 선택 알고리즘이 비해 가장 향상된 것을 볼 수 있었다. 그리고 Relief-F 알고리즘에 비해 시간복잡도에서 자질 값 차 계산부분을 최소 25% 이상 줄일 수 있었고, 자질을 필터링 후 tf*idf 문서행렬(Org)의 자질 발생 수에 비해 현저히 축소할 수 있고 정보이득과 Odds Ratio에 비해 자질 축소 비율이 전반적으로 안정적이었다. 이로 인해 좋은 분류 성능을 보였다. 향후 연구과제로 예제-기반 자질 필터링 알고리즘은 시간복잡도가 높으므로 대규모 크기 문서와 대규모 자질에 대한 예제-기반 자질 필터링 알고리즘에 대한 연구가 필요하다.

1) 자질 발생 수는 데이터셋 내 자질이 출현한 문헌 빈도의 합. 즉 자질 필터링 후 문서행렬 내에 남아 있는 자질 수의 합.

참 고 문 헌

[1] 이재윤, 최보영, 정영미, “문헌 자동분류에서 용어 가중치 기법에 대한 연구”, 한국정보관리학회 제7회 학술대회 논문집, pp.41-44, 2000.

[2] Yiming Yang and Jan O. Pederson, “A comparative study on feature selection in text categorization”, Proceedings of the 14th International Conference on Machine Learning ICML97, 1997, pp.412-420.

[3] Kira K & Rendell L, “A practical approach to feature selection”, Proceedings of the Ninth International Workshop on Machine Learning, Morgan Kaufmann Publishers Inc, 1992, pp.249-256.

[4] Igor Kononenko, “Estimating Attributes: Analysis and Extensions of RELIEF”, Proceedings of the 1994 European Conference on Machine Learning, 1994, pp.171-182.

[5] Yijun Sun, Jian Li, “Iterative RELIEF for feature weighting”, Proceedings of the 23rd international conference on Machine learning Vol.148, 2006, pp.913-920.

[6] Baranidharan Raman &Thomas R. Ioerger, “Instance based filter for feature selection”, Journal of Machine Learning Reseach 1, 2002, pp.1-23.

[7] Marko Robnik-Sikonja & Igor Kononenko, “Theoretical and Empirical Analysis of ReliefF and RReliefF”, Journal of Machine Learning Vol.53 Issue1-2, 2003, pp.23-69.

[8] Pascal Soucy & Guy W. Mineau, “A Simple KNN Algorithm for Text Categorization”, Proceedings of the 2001 IEEE International Conference on Data Mining, 2001, pp.647-648.

[9] Zhi-Hong Deng, Shi-Wei Tang, Dong-Qing Yang, Ming Zhang, Xiao-Bin Wu and Meng Yang, “Two Odds-Radio-Based Text Classification Algorithms”, Proceedings of Web Information Systems Engineering(Workshops) pp.223-231, 2002.

[10] Sanmay Das, “Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection”, The Proceedings of the Eighteenth International Conference on Machine Pages, pp.74-81, 2001.

박 흠

e-mail : parkheum2@empal.com

1988년 부산대학교 자연과학대학 계산통계학과(학사)

1998년 부산대학교 인지과학(이학석사)

2005년 부산대학교 정보통신공학(공학박사)

1988년~1990년 코닉시스템㈜

1990년~1998년 부산일보

현 재 부산대학교 인공지능연구실

관심분야 : 정보검색, 데이터마이닝, 자연어처리