Apriori - Genetic 알고리즘을 이용한 베이지안 자동 문서 분류

고 수 정† · 이 정 현††

요 약

기존의 베이지안 문서 분류는 문서의 특징 표현에 있어서 단어간의 의미를 정확하게 반영하지 못하는 문제점이 있다. 이러한 문제점을 해결하기 위해, 본 논문에서는 Apriori-Genetic 알고리즘을 이용한 베이지안 문서 분류 방법을 제안한다. Apriori 알고리즘은 단어간의 의미를 반영한 연관 단어의 형태로 문서의 특징을 추출하며 추출된 연관 단어로 연관 단어 지식베이스를 구축한다. Apriori 알고리즘만으로 연관 단어 지식베이스를 구축할 경우, 지식베이스 안에 부적당한 연관 단어가 포함된다. 따라서 문서 분류의 정확도가 낮아지는 단점이 있다. 이러한 단점을 보완하기 위해, Genetic 알고리즘을 이용하여 연관 단어 지식베이스를 최적화하는 방법을 사용한다. 베이지안 확률을 이용하는 분류자는 최적화된 연관 단어 지식베이스를 기반으로 문서를 클래스별로 분류한다. Apriori-Genetic 알고리즘을 이용한 베이지안 문서 분류의 성능을 평가하기 위해, Apriroi 알고리즘을 이용한 베이지안 문서 분류 방법, 역문헌빈도를 사용한 베이지안 문서 분류 방법, 기존의 단순 베이지안 분류 방법과 비교하였다.

Bayesian Automatic Document Categorization Using Apriori-Genetic Algorithm

Su-Jeong Ko† · Jung-Hyun Lee††

ABSTRACT

It is a problem that established Bayesian document categorization reflects the semantic relation inaccurately at feature expression of document. For the purpose of solving this problem, we propose Bayesian document categorizing method using Apriori-Genetic algorithm in this paper. Apriori algorithm extracts the feature of document being reflected semantics between words and constructs association word knowledge base through extracted association words. When association word knowledge base is constructed by Apriori algorithm, there are unsuitable association words in association word knowledge base. According to, it has a shortcoming that the accuracy of document categorization becomes lower. In order to complement a shortcoming, we use Genetic algorithm, which optimizes the association word knowledge base. Then, classifier using Bayesian probability categorizes documents based on optimized association word knowledge base. In order to evaluate performance of Bayesian document categorizing method using Apriroi-Genetic, we compare with Bayesian document categorizing method using Apriori algorithm and Bayesian document categorizing method using TFIDF and simple Bayesian document categorizing method.

키워드 : Apriori 알고리즘, 유전자 알고리즘, 베이지안 분류자, 문서 분류

1. 서 론

문서의 자동 분류에 대한 기존의 연구는 확률을 이용한 방법[7, 12], 통계적인 기법을 이용한 방법[4, 16], 벡터 유사도를 이용하는 방법[12] 등이 있다. 이들 중에서 베이지안 확률을 사용한 문서 분류는 각 언어권에서 일반적으로 높은 분류 효율을 나타내는 방법이다[11]. 기존의 단순 Naive Bayes 분류자[10]를 사용한 문서 분류는 학습문서와 실험문서에 출현한 모든 단어를 특징으로 추출하므로 문서의 특징을 정확히 반영하기 어렵다. 또한 모든 단어에 대해서 추정치를 계산하여 계산된 결과를 바탕으로 분류를 수행하므로 많은 잡음과 단어의 중의성이 분류에 영향을 끼친다. 이로 인한 문서의 오분류는 분류의 정확도를 저하시킨다. 이에 따라, 분류 정확도를 높이기 위해 역문헌빈도를 이용한 베이지안 문서 분류 방법[6, 15, 17]이 제안되었다. 제안된 방법은 학습문서와 실험문서에서 역문헌빈도를 사용하여 특징을 추출한다. 또한 실험문서에서 추출된 특징에 대해 가중치를 부여하므로 잡음으로 인한 오분류는 단순 Naive Bayes 분류자를 사용한 방법보다 줄어든다. 그러나 추출된 문서의 특징이 단어간의 의미 관계를 반영하지 못

† 정 회 원 : 인하대학교 대학원 전자계산학과
†† 종신회원 : 인하대학교 전자계산공학과 교수
논문접수 : 2001년 2월 20일, 심사완료 : 2001년 5월 4일

하므로 단어 의미 중의성 문제로 인해 문서가 오분류되는 문제점은 해결하지 못하였다.

이를 해결하기 위해, 본 논문에서는 Apriori-Genetic 알고리즘을 이용한 베이지안 문서 분류 방법을 제안한다. 제안된 방법에서 Apriori 알고리즘은 학습문서와 실험문서의 특징을 단어간의 의미를 반영한 연관 단어 형태로 추출한다. 이에 따라 단어 의미 중의성 문제로 인한 문서 분류의 오분류를 해결하므로 문서 분류의 정확도가 높아진다. 또한 잡음으로 인한 분류의 오류를 줄이기 위해 학습문서에서 추출된 연관 단어로 연관 단어 지식베이스를 구축하고 유전자(Genetic)알고리즘을 이용하여 이를 최적화한다. Naive Bayes 분류자는 연관 단어 지식베이스의 연관 단어에 추정치를 부여하고 연관 단어의 형태로 특징이 추출된 실험문서를 클래스로 분류한다. 이러한 방법으로 제안된 베이지안 문서 분류 방법의 성능을 평가하기 위해, Apriori 알고리즘을 이용한 베이지안 문서 분류 방법, 역문헌빈도를 이용한 베이지안 문서 분류 방법, 기존의 단순 베이지안 분류 방법과 비교하였다.

2. 기존의 Naive Bayes 분류자를 이용한 문서 분류

텍스트 문서의 분류를 위한 대부분의 연구는 Naive Bayes 분류자라고 불리는 변형된 베이지안 분류법을 사용하였다[15]. 단순 Naive Bayes 분류자는 문서에 나타나는 모든 단어를 특징으로 추출한다. Naive Bayes 분류자는 실험문서(D)의 특징이 $\{n_1, n_2, \cdots, n_k, \cdots, n_m\}$라고 하였을 경우 식 (1)에 의해 {class1, class2, ···, classID, ···, classN} 중 하나의 클래스로 실험문서(D)를 분류한다.

$$class = \arg \max_{classID=1}^{N} P(classID) \prod_{k=1}^{m} P(n_k \mid classID) \quad (1)$$

식 (1)에서 P(classID)는 classID로 분류될 확률이며, $P(n_k|classID)$는 n_k가 classID에 있을 추정치이다. $\{n_1, n_2, \cdots, n_k, \cdots, n_m\}$의 각 단어는 문맥에 관계 없이 독립적이라고 가정한다. 독립 가정을 전제로 하는 각 단어에 대한 $P(n_k|classID)$의 확률은 식 (2)에 대입함으로써 구한다.

$$p(W_k \mid classID) = \frac{num_{k\,classID} + 1}{num_{classID} + |Voc|} \quad (2)$$

식 (2)에서 $num_{classID}$는 classID 내의 단어의 총 개수이며, $numk_{classID}$는 classID에서 단어 w_k의 출현 빈도수, 그리고 |Voc|는 classID의 총 어휘수이다.

역문헌빈도를 이용한 Naive Bayes 분류자는 문서의 특징을 추출하기 위해 문서를 형태소 분석하고 그 결과 중에서 명사만을 추출한다. 추출된 모든 명사의 역문헌빈도[14]는 식 (3)을 이용하여 구한다.

$$W_{nk} = f_{nk} \cdot [\log_2 \frac{n}{DF} + 1] \quad (3)$$

fn_k는 문서 내 모든 단어에 대한 단어 n_k의 상대빈도이며, n은 학습문서의 수이며 DF는 학습문서에서 단어 n_k가 나타난 문서의 수를 의미한다. 역문헌빈도가 높은 단어부터 낮은 단어로 정렬하여 상위의 빈도를 나타내는 단어만을 문서의 특징으로 추출한다. Naive Bayes 분류자는 실험문서(D)의 특징이 $\{n_1, n_2, \cdots, n_k, \cdots, n_m\}$라고 하였을 경우 식 (4)에 의해 {class1, class2, ···, classID, ···, classN} 중 하나의 클래스로 실험문서(D)를 분류한다.

$$class = \arg \max_{classID=1}^{N} P(classID) \prod_{k=1}^{m} W_{nk} P(n_k \mid classID) \quad (4)$$

식 (4)는 특징에 가중치를 부여한 점에서 식 (1)과 다르다. 여기서의 가중치는 식 (3)에 의해 구해진 역문헌빈도를 의미한다. $P(n_k|classID)$의 확률은 단순 Naive Bayes 분류자와 같이 식 (2)를 이용한다.

3. Apriori-Genetic 알고리즘을 이용한 베이지안 자동 문서 분류

3.1 문서의 특징 표현

본 논문에서는 텍스트로 이루어진 문서를 표현하기 위해 형태소 분석을 통한 명사 추출 과정을 전처리 과정으로 사용한다. 전처리 과정을 통하여 추출된 명사들을 대상으로 연관 단어를 마이닝한 결과, 각 문서는 연관 단어들의 집합, 즉 연관 단어 벡터 모델로 나타내어진다. 본 논문에서 사용한 형태소 분석 시스템은 사용자 중심의 지능형 정보검색 시스템[13]에서 사용된 방법을 사용하여 연관 단어 벡터 모델은 형태소 분석의 복잡한 부분인 파싱(parsing)을 통한 의미 분석을 생략하여 추출된 명사만을 사용한다. Apriori 알고리즘[1, 2]은 형태소 분석에 의해 추출된 명사들로부터 연관 단어를 마이닝한다. 그 결과, 문서는 $\{(w_{11} \& w_{12} \cdots \& w_{1(r-1)} => w_{1r}), (w_{21} \& w_{22} \cdots \& w_{2(r-1)} => w_{2r}), \cdots, (w_{k1} \& w_{k2} \cdots \& w_{k(r-1)} => w_{kr}), \cdots, (w_{m1} \& w_{m2} \cdots \& w_{m(r-1)} => w_{mr})\}$ 형태의 연관 단어 벡터 모델로 표현된다. 여기서, $(w_{11} \& w_{12} \cdots \& w_{1(r-1)} => w_{1r})$는 연관 단어를 나타내며, $\{w_{11}, w_1, w_{1(r-1)}, w_{1r}\}$는 연관 단어를 구성하는 단어들의 구성이며, r은 연관 단어를 구성하는 단어의 수이다. 또한, m은 텍스트를 대표하는 연관 단어의 수이다.

3.2 연관 단어 지식베이스를 위한 연관 단어의 마이닝

연관 단어는 Apriori 알고리즘[1, 2]에 의해 마이닝되며, 마이닝된 결과는 연관 단어 지식 베이스에 저장된다. 본 절에서는 연관 규칙 마이닝에 사용되는 Apriori 알고리즘이 연관 규칙을 이용하여 연관 단어를 마이닝하도록 하는 방

법을 설명한다.

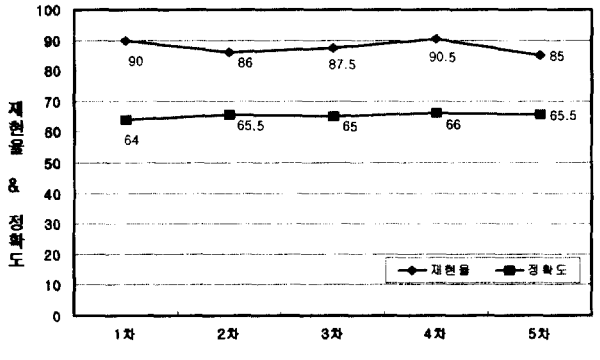
신뢰도를 결정하기 위한 식 (5)는 다음과 같이 구해진다. 식 (5)는 단어W1과 W2의 모든 항목을 포함하고 있는 트랜잭션의 수를 단어 W1의 항목을 포함하고 있는 트랜잭션의 수로 나눈 결과 값을 나타낸다.

$$Confidence(W1->W2) = Pr(W2|W1) \quad (5)$$

지지도를 결정하기 위한 식 (6)은 전체 단어들의 쌍 중에 각 연관 단어의 출현 빈도를 나타낸다. 식 (6)은 단어 W1과 W2의 모든 항목을 포함하고 있는 트랜잭션의 수를 데이터베이스 내의 전체 트랜잭션의 수로 나눈 결과 값을 나타낸다.

$$Support(W1->W2) = Pr(W1 \cup W2) \quad (6)$$

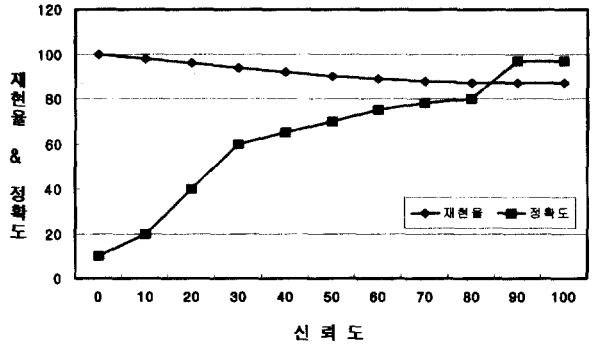
Apriori 알고리즘의 신뢰도(confidence)와 지지도(support)에 따라 마이닝되는 연관 단어 쌍의 목록은 현저하게 차이를 보인다. 본 논문에서는 연관 단어 쌍을 구성하기 위한 신뢰도와 지지도를 결정하기 위하여 100개의 웹문서를 대상으로 신뢰도와 지지도에 따라 마이닝되는 연관 단어의 재현율과 정확도를 살펴보았다. 100개의 웹문서는 본 논문이 실험을 위해 컴퓨터 분야의 웹문서를 8개의 클래스로 분류한 클래스 중에서 게임 클래스에 수집된 웹문서이다. (그림 1)은 수집된 웹문서에 따른 연관 단어의 재현율과 정확도의 차이를 보이기 위한 그림이다. 웹문서는 게임 클래스로부터 웹문서 수집기를 이용하여 1차에 100개씩 총 5차에 걸쳐서 수집된다. 1차에서 5차까지 연관 단어를 마이닝할 때 사용되는 신뢰도와 지지도는 0~100사이의 값 중에서 중간 값인 50으로 동일하게 지정하였다. (그림 1)에서 1차에서 5차까지의 재현율과 정확도는 각각의 평균을 기준으로 최대 3의 차이를 보인다. 이에 따라 본 논문에서는 평균에 가장 근사한 재현율과 정확도를 나타낸 3차에서 수집한 웹문서를 대상으로 신뢰도와 지지도의 변화에 따른 연관 단어의 재현율과 정확도를 조사하였다. 재현율과 정확도를 평가하는 기준은 영어 단어에 대한 시소러스인 WordNet[5]을 사용하여 평가하였다.



(그림 1) 수집된 웹문서에 따른 연관 단어의 재현율과 정확도

평가를 위해 WordNet에서 게임과 관련된 영어 단어의 동의어, 상의어, 하의어를 추출하였다. 추출된 단어를 한글로 번역하여 300개의 연관 단어를 구성하였다. 마이닝된 연관 단어가 이들 300개의 연관 단어에 포함되지 않을 경우 오류로 처리하였다. 정확도는 마이닝된 연관 단어 중에서 오류로 처리된 연관 단어의 비율을 나타낸다. 재현율은 마이닝된 연관 단어가 평가를 위해 구성된 연관 단어에 포함된 비율이다.

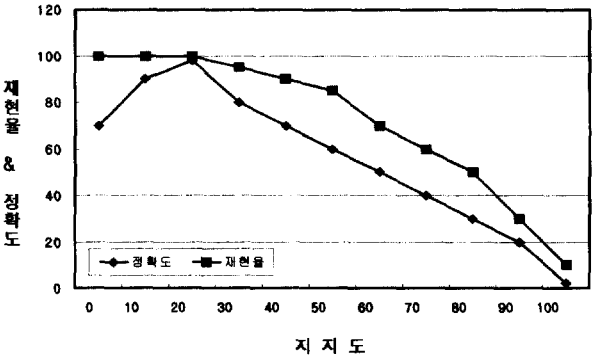
(그림 2)는 100개의 웹문서를 대상으로 신뢰도를 다양하게 변화시켰을 때, 추출된 연관 단어에 대한 정확도와 재현율을 나타낸다.



(그림 2) 신뢰도의 변화에 따른 재현율과 정확도

(그림 2)는 신뢰도가 클수록 추출되는 연관 단어의 정확도는 높아지나 재현율은 낮아짐을 나타낸다. 그러나 85이상의 신뢰도에서는 재현율이 거의 일정하고 정확도는 높은 수치를 나타낸다. 따라서 가장 적합한 연관 단어를 추출하기 위해서는 신뢰도를 85이상으로 지정해야 한다.

(그림 3)은 100개의 웹문서를 대상으로 지지도를 다양하게 변경시킴에 따른 정확도와 재현율의 변화를 나타낸다. 정확도와 재현율의 측정 기준은 신뢰도와 같다.



(그림 3) 지지도의 변화에 따른 재현율과 정확도

(그림 3)에서 정확도와 재현율의 곡선이 일치하는 지점은 지지도가 22인 경우로, 이 지점에서 가장 적합한 연관 단어가 추출된다. 그러나, 지지도가 22이상인 경우에는 정확도와 재현율이 모두 낮아진다. 따라서 가장 신뢰할 만한 연관 단어를 추출하기 위해서는 22이하의 지지도를 지정해

야 한다. 그러나 지지도를 0으로 한다면 클래스와 전혀 관계 없는 문서에서 연관 단어가 추출되므로 0보다 크도록 설정하여야 한다.

<표 1>은 Apriori 알고리즘을 사용하여 게임 클래스의 학습문서 중 100개 문서를 대상으로 지지도 20, 신뢰도 90으로 지정하였을 경우 마이닝된 연관 단어의 예이다. 이들 연관 단어는 연관 단어 지식베이스에 저장된다.

〈표 1〉 연관 단어 지식베이스의 연관 단어(게임 클래스)

(1)게임&구성&선수&경기&스포츠&참가=>선발	(10)게임&이용&문제=>규칙
(2)국내&최신&기술&설치=>개발	(11)그림&인기&서비스=>음악
(3)게임&참가&인기&사용자&접속=>이벤트	(12)그림&데이터&서비스=>엔진
(4)운영&선발&경기&순위&규칙=>평가	(13)데이터&프로그램=>음악
(5)게임&순위&이름=>스포츠	(14)그림&데이터&프로그램=>사진
(6)운영&스포츠&위원회&선수=>선발	(15)게임&설명&제공=>공략
(7)게임&구성&선발&순위=>경기	(16)게임&이용&기술=>개발
(8)게임&일정&선수&참가&운영=>스포츠	(17)삭제&게임&개인전=>경고
(9)데이터&암호&통신망=>가입	(18)게임&제공&일러스트=>설명

3.3 연관 단어 지식베이스의 최적화를 위한 유전자 (Genetic) 알고리즘

Apriori 알고리즘을 이용하여 구축된 연관 단어 지식베이스는 단어간의 의미 관계를 표현하므로 단어의 중의성 문제를 해결할 수 있다. 그러나 이러한 지식베이스는 클래스에 적합하지 않은 연관 단어를 포함하므로 이를 최적화해야 한다. 본 절에서는 유전자 알고리즘이 연관 단어 지식베이스의 최적화에 적용되기 위한 지식표현 방법 및 절차를 설명한다.

유전자 알고리즘은 유전인자(gene), 염색체(chromosome), 모집단(population)을 사용하여 연관 단어 지식베이스를 최적화한다. 본 논문에서의 유전인자는 Apriori 알고리즘에 의해 추출된 연관 단어를 표현하기 위해 0과 1의 값을 갖는 비트로 표현된다. 염색체는 학습될 문서를 의미하며, 비트로 표현되는 유전인자의 집합으로 구성된다. 모집단은 실험대상의 문서로써 추출한 전체 문서를 나타낸다. 모집단은 초기화, 적합율 계산, 재구성, 선택, 교배, 돌연변이, 평가의 과정을 통하여 진화한다. 진화된 모집단은 평가를 통하여 다음 세대로 진화를 계속할 것인가가 결정된다.

초기화 단계에서는 학습문서를 염색체로 정의하고 유전인자로 표현한다. 이를 위해 학습문서를 형태소 분석한다. <표 2>는 개임에 관한 10개의 문서를 형태소 분석한 예이다.

〈표 2〉 모집단으로 선택한 문서의 명사 추출

	형태소 분석 결과에 의해 추출된 명사들
1	구성,페이지,창세기,규칙,이미지,배경,공략,갤러리,링크,설명,아이템,페이지,에피소드,일러스트
2	참가,결혼,선수,커플,갤러리,공략,배경,설명,수록,시리즈,시스템,에피소드,이미지,인물..
3	개인전,개편,게임,경고,경기,규칙,그룹,기준,답변,대전,대표,대림,문제,방법,삭제,스포츠,신청,..
4	기술,규칙,선수,공략,인기,접속,사용자,파일,개발,참가,운영,순위,평가,경기,구성,설명,게임,선발
5	리스트,게임,버튼,대전,이벤트,컴퓨터,하드웨어,정보,제공,구매,사용,대결,회원,등록,암호,유료,.
6	게임,이용,규칙,설명,제공,인기,사용자,선수,참가,운영,일정,문제,제공,공략,기술,경기,순위,선발
7	스포츠,야구,용병,위치,경기,구성,기록,적용,실전,자동,제품,대회,진행,참가,참전,그룹,변경
8	게임,국내,참가,인기,접속,작업,기술,설치,운영,순위,이름,선수,선발,경기,이용,복구,정보
9	멀티미디어,사운드,국내,제공,제작,소프트웨어,음악,데이터,프로그램,그림,서비스,엔진,게임,,,
10	개인전,스포츠,단체전,프로,아마,신청,대회,참가,우승,준우승,토너먼트,답변,회원,개편

<표 1>의 연관 단어에 포함된 명사가 <표 2>의 명사에 존재한다면 염색체를 구성하기 위한 유전인자는 1의 값을 갖는다. 존재하지 않는다면 유전인자는 0의 값을 갖는다. <표 3>은 이러한 방식으로 염색체를 표현한다. 또한 모집단은 구성된 염색체의 모임이다.

〈표 3〉 문서의 염색체 표현

문서	염 색 체
1	1011000000000100000
2	0010000000000001001
3	1010111010000011111
4	1111111010000011111
5	1010101010000011111
6	1111111010000011111
7	1000111100000000000
8	1111111010000011111
9	1111101101011111111
10	1010110100000000010

유전자 알고리즘에서는 개체의 성능을 다른 개체와 비교하기 위하여, 혹은 개체가 얼마나 유전자 알고리즘이 적용되고 있는 가상의 환경에 잘 적응하고 있는지를 나타내는 척도로서 적합율을 모든 개체에 부여한다. 본 논문에서 선택한 적합율의 기준은 선택한 문서 간에 유사도가 높다면 적합율이 높은 것으로 유사도가 낮다면 적합율이 낮은 것으로 판정하였다. 문서 간의 유사도[9]는 비트간의 일치 수를 사용하여 유사도를 구해야 하므로 식 (7)의 Jaccard 방법[3]을 이용한다. 식 (7)에서 #(doc*n* ∪ doc*m*)은 문서n(doc*n*)을 표현한 염색체에서 1의 값을 갖는 유전인자와 문서m(doc*m*)을 표현한 염색체에서 1의 값을 갖는 유전인자의 합집합의 수를 의미하며, #(doc*n* ∩ doc*m*)은 문서n(doc*n*)을 표현한 염색체에서 1의 값을 갖는 유전인자와 문서m(doc*m*)을 표현한 염색체에서 1의 값을 갖는 유전인자의 공집합의 수를 의미한다.

$$\text{Fitness}(\text{doc}n,\text{doc}m) = \#(\text{doc}n \cap \text{doc}m)/\#(\text{doc}n \cup \text{doc}m) \quad (7)$$

식 (7)을 이용하여 클래스에 포함된 각 염색체의 적합율을 구할 수 있으며, 또한 클래스에 포함된 전체 문서의 평균 적합율을 구할 수 있다.

재구성 단계에서는 적합율 계산을 위한 목적 함수의 값을 다른 값으로 재구성한다. 재구성하는 목적은 적합율을 선택연산자를 적용하는 확률로 사용하기 위함이다. 이러한 작업을 적합율 조정(fitness scaling)이라고 한다. 본 논문에서는 적합율의 조정을 위해 식 (8)을 이용한다. 식 (8)은 클래스에 속한 전체 문서의 적합율의 합에 대한 각 문서 적합율(Fitness [class, doc])의 비를 계산한다. 이에 따라 적합율이 재조정되어 Fitness-s[class,doc]에 저장된다. Fitness[class, doc]는 클래스에 속한 문서의 적합율을 저장한 배열이며 Fitness-s[class, doc]는 재조정된 적합율을 저장하는 배열이다.

$$\text{Fitness\_s}[class.doc] = \frac{\text{Fitness}[cl\ ass.doc]}{\sum_{doc=1}^{r} \text{Fitness}[cl\ ass.doc]} \quad (8)$$

<표 4>는 식 (7)를 이용하여 <표 3>의 각 염색체의 적합율을 구하고 식 (8)를 이용하여 적합율을 조정한 결과를 나타낸다.

〈표 4〉 염색체의 적합율 및 재조정 적합율

문 서	적 합 율	재조정 적합율
문서1	0.277170	5.354191
문서2	0.275670	5.325215
문서3	0.649913	12.554600
문서4	0.612508	11.832030
문서5	0.616830	11.915520
문서6	0.665659	12.858770
문서7	0.391846	7.569429
문서8	0.665659	12.858770
문서9	0.557083	10.761370
문서10	0.464354	8.970091
평균	0.5176692	10

선택 단계에서는 재구성된 적합율을 바탕으로 교배할 문서를 선택한다. 교배 단계에서는 선택된 문서를 대상으로 교배를 한다. 교배 방법에는 1점 교차방법, 2점 교차방법, n점 교차방법, 균일 교차방법 등이 있으나 본 논문에서는 1점 교차방법을 사용한다. 1점 교차 방법은 임의의 한 지점을 선택하여 그 이후의 유전인자를 다른 문서의 유전인자와 맞바꾸는 형식으로 동작한다. 예를 들어, 18개의 유전인자를 갖는 개체일 경우 임의의 한 지점이란 [1..18]사이에서 발생한 임의의 지점이다. 교배를 하기 위해서는 교배율을 지정해야 한다. 교배율이란 선택된 문서가 교배되는 확률을 의미한다. 통상적으로 교배율(Pc)은 0.7-0.9를 사용하며, 본 논문에서는 0.9를 사용하였다. 교배율을 적용하는 방식은 선택된 부모 염색체에 [0‥1]사이의 임의의 수(r)를 적용하여 r<=Pc이라면 선택된 부모 염색체는 교배되며, r>Pc이라면 선택된 부모 염색체는 교배되지 않고 부모 염색체가 바로 자손이 된다.

돌연변이 단계에서는 하나의 비트를 주어진 확률에 따라서 다른 값으로 바꾸는 단계이다. 돌연변이율은 염색체의 유전인자가 다른 값으로 바뀌는 확률을 의미한다. 돌연변이율은 0.01-0.05사이의 값을 많이 사용하며 본 논문에서는 돌연변이율을 0.01로 정한다. <표 5>는 <표 3>의 부모 염색체가 선택, 교배, 돌연변이 단계를 거쳐 탄생된 2세대의 염색체이다.

〈표 5〉 2세대 염색체 및 적합율

문서	염색체(2세대)	적합율
문서1	1010111111000011111	0.700600
문서2	1111111101000001111	0.716735
문서3	1010001101000011111	0.629429
문서4	1010110101000011111	0.693230
문서5	1101111101000000000	0.525678
문서6	1000111101000011111	0.700791
문서7	1101111101011111111	0.625147
문서 8	1111101001000011111	0.646749
문서 9	1111111101100010101	0.625490
문서10	1010110100010000111	0.567764
평균		0.643161

평가 단계에서는 진화를 계속할 것인가 종료할 것인가를 결정하는 단계이다. 평가하는 기준은 평균 적합율이 적합율 임계값과 같거나 크다면 진화를 종료하고 작다면 재구성 단계부터 다시 진화를 반복한다. 본 논문에서는 적합율 임계값을 1로 하여 계산된 평균 적합율이 1보다 작다면 진화를 계속 진행한다. <표 5>의 3열에 계산된 2세대 염색체의 평균 적합율 0.643161이므로 다음 세대로의 진화가 계속된다.

<표 6>은 1세대에서 마지막 세대인 8세대까지의 염색체와 각 세대의 평균 적합율을 보인다.

〈표 6〉 염색체의 진화

진화	염 색 체	적 합 율	평균 적합율
1 세대	문서1:1011000000000100000 문서2:0010000000000001001 .	0.277170 0.275670 .	0.517669
2 세대	문서1:1010111111000011111 문서2:1111111101000000111 .	0.700600 0.716735 .	0.643161
3 세대	문서1:1010111101000011111 문서2:1111111101000011111 ..	0.774610 0.774650 .	0.684610
4 세대	문서1:1010111111000011111 문서2:1111111100000011111 .	0.790659 0.736813 .	0.8215736
5 세대	문서1:1000111101000011111 문서2:1010111111000011011 .	0.834790 0.768407 .	0.8215740
6 세대	문서1:1011111110000011111 문서2:1110111110000011011 .	0.779121 0.718132 .	0.8597770
7 세대	문서1:1010111101000011111 문서2:1010111101000011111 .	0.991667 0.991667 .	0.9850000
8 세대	문서1:1010111101000011111 문서2:1010111101000011111 문서3:1010111101000011111 문서4:1010111101000011111 문서5:1010111101000011111 문서6:1010111101000011111 문서7:1010111101000011111 문서8:1010111101000011111 문서9:1010111101000011111 문서10:1010111101000011111	1 1 1 1 1 1 1 1 1 1	1.0000000

진화가 진행되는 동안 <표 6>의 3열에서와 같이 평균 적합율은 일정하게 증가되어 평균 적합율이 1이 되는 8세대에서 진화는 종료되었다. 8세대의 염색체에서 1의 유전인자를 나타내는 연관 단어는 채택되고 0의 유전인자를 나타내는 연관 단어는 제거된다. 이에 따라 <표 1>에 나타난 게임 클래스의 연관 단어 중에서 0의 유전인자를 나타내는 연관 단어를 제거하면 연관 단어 지식베이스 내 게임 클래스는 최적화된다.

3.4 최적화된 연관 단어 지식베이스에 기반한 Naive Bayes 분류자

Naive Bayes 분류자는 학습 단계와 분류 단계를 통하여 문서를 분류할 수 있다. 학습 단계에서는 Apriori와 유전자

알고리즘에 의해 구축된 연관 단어 지식베이스의 연관 단어에 추정치를 부여한다. 연관 단어 지식베이스의 *classID*에 있는 k번째 연관 단어 (w_{k1} & w_{k2} ⋯ & $w_{k(r-1)}$ => w_{kr})에게 추정치를 부여하기 위해서 식 (9)를 이용한다. 본 논문에서는 *classID*에 있는 k번째 연관 단어 (w_{k1} & w_{k2} & ⋯ $w_{k(r-1)}$ => w_{kr}) 의 추정치는 P((w_{k1} & w_{k2} ⋯ & $w_{k(r-1)}$ => w_{kr}) |*classID*)로 표현한다. 여기서, n은 학습문서로부터 마이닝된 연관 단어의 전체 수이고, n_k는 전체 개수n 중에서 지식베이스에 있는 연관 단어 (w_{k1} & w_{k2} ⋯ & $w_{k(r-1)}$ => w_{kr})와 일치하는 연관 단어의 수이다. 또한, *classID*는 연관 단어 지식베이스에 있는 클래스의 레이블이며, |AWKB|는 연관 단어 지식베이스에 있는 전체 연관 단어의 수이다.

$$P((w_{k1}\ w_{k2}\ \cdots\ w_{k(r-1)} \Rightarrow w_{kr}) \mid classID) = \frac{n_k + 1}{n + \mid AWKB \mid} \quad (9)$$

학습 과정은 누적 단계와 추정치 부여 단계로 나눈다. 누적 단계에서는 학습문서에 있는 연관 단어가 지식베이스 안에 있는 경우 횟수가 누적된다. 가중치 부여 단계에서는 누적 단계의 결과를 식 (9)에 적용하여 지식베이스의 연관 단어에 추정치를 부여한다. 이러한 과정을 통해, 지식베이스의 연관 단어에 추정치가 추가된다. <표 7>은 <표 1>의 연관 단어 지식베이스와 최적화된 연관 단어 지식베이스에 각각 식 (9)를 사용하여 추정치를 추가한 결과이다. AWKB는 연관 단어 지식베이스를 의미하며, OAWKB는 최적화된 연관 단어 지식베이스를 의미한다.

〈표 7〉 게임 클래스의 연관 단어에 추정치 부여

연관 단어	n_k	n	|AWKB| (|OAWKB|)	P(w_{k1}&w_{k2}...&$w_{k(r-1)}$) =>w_{kr}|game class) (AWKB)	P(w_{k1}&w_{k2}...&$w_{k(r-1)}$) =>w_{kr}|game class) (OAWKB)
게임&구성&선수&경기&스포츠&참가=>선발	23	25	231(161)	0.09375	0.129032
국내&최신&기술&설치=>개발	2	5	231(161)	0.0127	Delete
게임&참가&인기&사용자&검색=>이벤트	25	28	231(161)	0.100386	0.137566
운영&선발&경기&순위&규칙=>평가	3	6	231(161)	0.016878	Delete
게임&순위&이름=>스포츠	22	26	231(161)	0.089494	0.122994
운영&스포츠&위원회&선수=>선발	25	28	231(161)	0.100386	0.137566
게임&구성&선발&순위=>경기	21	23	231(161)	0.086614	0.119665
게임&일정&선수&참가&운영=>스포츠	23	27	231(161)	0.093023	0.127659
데이터&암호&통신망=>가입	1	4	231(161)	0.008511	Delete
게임&이용&문제=>규칙	21	25	231(161)	0.085938	0.118279
그림&인기&서비스=>음악	1	3	231(161)	0.008547	Delete
그림&데이터&서비스=>엔진	3	4	231(161)	0.017021	Delete
데이터&프로그램=>음악	4	5	231(161)	0.021186	Delete
그림&데이터&프로그램=>사진	5	6	231(161)	0.025316	Delete
게임&소프트=>버전	24	29	231(161)	0.096154	0.1315789
게임&이용&기술=>개발	25	28	231(161)	0.100386	0.137566
삭제&게임&개인전=>경고	25	27	231(161)	0.100775	0.138297
게임&이미지&인상=>설명	21	26	231(161)	0.085603	0.117647

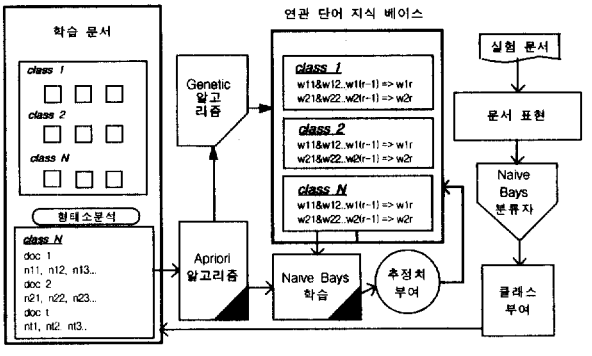
분류 단계에서는 추정치가 부여된 연관 단어 지식베이스를 사용하여 Naive Bayes 분류자에 의해 실험문서를 클래스로 분류할 수 있다. Apriori 알고리즘은 3.2절에서 기술한 방법으로 실험문서로부터 연관 단어의 형태로 특징을 추출한다. 실험문서(D)의 특징이 {d(w_{11} & w_{12} ⋯& $w_{1(r-1)}$ = > w_{1r}), d(w_{21} & w_{22} ⋯& $w_{2(r-1)}$ = > w_{2r}), ⋯, d(w_{k1} &w_{k2} ⋯ & $w_{k(r-1)}$ => w_{kr}), ⋯, d (w_{m1} & w_{m2} ⋯ & $w_{m(r-1)}$ = > w_{mr})}라고 하였을 때 Naive Bayes 분류자는 식 (10)을 이용하여 실험문서를 클래스로 분류한다. d(w_{k1} & w_{k2} ⋯ & $w_{k(r-1)}$ => w_{kr})의 "d"는 실험문서로부터 추출된 연관 단어임을 나타낸다.

$$class = \arg\max_{classID=1}^{N} P(classID) \prod_{k=1}^{m} P(d(w_{k1}\ w_{k2}\ \cdots\ w_{k+1} \Rightarrow w_k) \mid classID) \quad (10)$$

식 (10)에서 문서D가 분류될 클래스는 *class*이며, 전체 클래스의 수는 N이다. 또한 P(d (w_{k1} & w_{k2} ⋯ & $w_{k(r-1)}$ = > w_{kr}) | *classID*)는 식 (9)에 의한 추정치이며 P(classID)는 classID로 분류될 확률이다.

4. 전체 시스템 설계 및 베이지안 문서 분류의 예

이 장에서는 전체 시스템 설계도에 따라 최적화된 연관 단어 지식베이스를 구축하며, 이를 기반으로 문서를 분류하는 방법을 구체적으로 설명한다. (그림 4)는 본 논문에서 설계한 베이지안 자동 문서 분류를 위한 시스템의 구성도를 나타낸다.



(그림 4) Apriroi-Genetic을 이용한 베이지안 문서 분류의 구성도

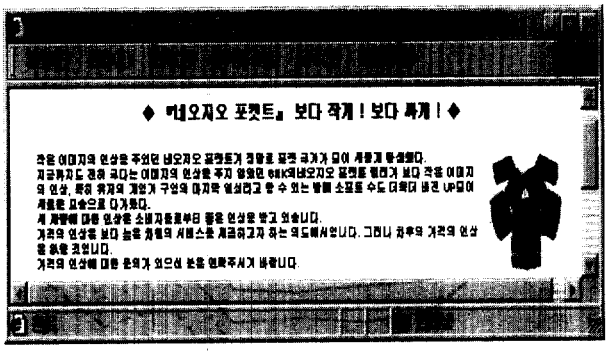
4.1 학습문서 및 실험문서

학습문서는 한국어 정보검색 시스템의 성능 평가용 데이터 집합인 KTset95 문서 4,414개 중 2400개의 문서로, 실험문서는 웹문서 수집기에 의해 컴퓨터 분야의 URL로부터 수집한 800개의 웹문서와 KTset95 문서 중 800개의 문서를 병합하여 구성한다. 학습문서의 클래스는 전산학 각 연구분야의 8개 클래스로 수작업으로 분류하였다. 학습문서와 실험문서의 실험 대상을 다르게 설정한 이유는 본 논문에서 제시한 방법에 대한 정확한 평가를 위함이다. 여기서 8개의 클래스는 {게임, 그래픽, 뉴스와 미디어, 반도체, 보안, 인터넷, 전자출판, 하드웨어}의 레이블이며 (그림 4)에서 {class1, class2, ⋯, classN}으로 표현된다. 이렇게 8개의 클래스로 분류한 기준은 알타비스타, 야후, 한미르 등의 기존의 검색 엔진이 컴퓨터 분야의 주제를 대상으로 분류한 통계에 따른 것이다. 따라서 각 클래스에 300개의 문서가 학

습문서로 할당된다. KTset95 문서 중 정의된 클래스에 해당하지 않는 문서들은 사용하지 않았다. (그림 4)의 학습문서에서 {doc 1, doc 2, …, doc t}는 훈련을 위해 클래스로 분류한 문서를 의미한다. 한 클래스에 300개의 문서가 속하게 되므로, t는 300의 값을 나타낸다. {nt1, nt2, nt3, …}은 문서 doc t를 대상으로 형태소 분석한 결과 추출된 명사를 의미한다.

4.2 Naive Bayes 분류자에 의한 실험문서 분류의 예

본 절에서는 기존의 Naive Bayes 분류자와 본 논문에서 제안한 Apriori_Genetic 알고리즘을 이용한 Naive Bayes 분류자가 웹문서를 분류하는 예를 보인다. (그림 5)는 Naive Bayes 분류자가 분류하기 위한 웹문서이다. (그림 5)의 웹문서는 게임기에 대한 홍보를 하는 홈페이지이므로 게임클래스의 class1에 분류되어야 한다.



(그림 5) http : //nihongobank.co.kr/ichinichi/1999/19991025 h.htm 의 URL에 나타난 웹문서

(그림 6)은 (그림 5)의 웹문서를 형태소 분석한 결과의 일부와 분석 결과 추출된 명사와 출현 횟수를 보인다.

(형태소 분석 결과)
[네오지오 ((네 오지 오) (N N N))] [포켓트] [보다] [작게 ((작 게) (A EC))] [보다] [싸게 ((싸 게) (V EC))] [작은 ((작 으 ㄴ) (A EU ED))] [이미지의 ((이미지 의) (N PCD))] [인상을 ((인상 을) (N PCO))] [주었던 ((주 었 던) (N EPF ED)) ((주 었 던) (V EPF ED)) ((주 었 던) (VA EPF ED))] [네오지오 ((네 오지 오) (N N N))] [포켓트가] [정말로 ((정말 로) (N PCA))] [포켓 ((포켓) (N))]....
(추출된 명사 및 출현 횟수)
인상(8),이미지(3),오지(3),가격(3),바탕(2),포켓(1)게임(1),차원(1),소프트(1),버전(1),열쇠(1),소비자(1),제품(1),수정(1),구입(1),대한(1),서비스(1),연락(1),유저(1),컬러(1),문의(1),수도(1),주지(1),제공(1),주시(1),차후(1),모습(1),발매(1),정말(1),지금(1)

(그림 6) 형태소 분석 결과 및 추출된 명사

<표 8>은 (그림 6)에 나타난 명사와 출현횟수를 사용하여 본 논문에서 Apriori-Genetic 알고리즘을 이용한 방법과 기존의 방법으로 Naïve Bayes 분류자가 (그림 5)의 문서를 분류하는 예를 보인다. 단순 Naïve Bayes 분류자는 (그림 6)의 모든 명사를 대상으로 추정어를 산출하므로 각 클래스별로 많은 잡음이 들어간다. 예를 들어 "제공"이라는 단어는 문서에서 중요한 단어라고 할 수 없으나 class1, class2, class8에서 각각의 추정치를 산출하여 식 (10)의 문서 분류식에 대입되므로 문서 분류의 정확도를 저하시킨다. 또한 Naïve Bayes 분류자는 <표 9>의 "인상"이라는 단어에 대해 "가격 인상"의 인상을 의미하는지 이미지의 인상의 인상을 의미하는지 판별할 수 없으므로 "인상"이라는 단어의 의미는 다르나 같은 단어로 취급된다. 이러한 원인으로 인하여 단순 Naïve Bayes 분류자는 (그림 5)의 웹문서를 class2로 오분류한다.

역문헌빈도를 이용한 Naïve Bayes 분류자는 (그림 6)의 형태소 분석 결과 나온 단어에 대해 식 (3)을 이용하여 단어에 대한 역문헌빈도를 계산한다. 계산 결과, 역문헌빈도가 높은 단어부터 정렬한 후 상위 단어만을 문서의 특징으로 채택한다. Naïve Bayes 분류자는 역문헌빈도를 가중치로 설정하고 식 (4)를 이용하여 문서를 분류한다. 역문헌빈도를 이용한 Naïve Bayes 분류자는 가중치가 높은 단어만을 주제어로 채택함에 의해 가중치가 낮은 단어로 인해 발생하는 문서 분류의 오류는 감소하였으나 단어의 중의성 문제는 문서 분류의 정확도를 저하시킨다. 단순 Naïve Bayes 분류자의 예와 같이 "인상"이라는 의미 중의성이 발생되므로 "이미지"의 인상에 대한 역문헌빈도가 가장 높으므로 class2의 그래픽 클래스에 오분류된다.

〈표 8〉 Naïve Bayes 분류자가 실험문서를 분류하는 예

Naïve Bayes 분류자	특징 추출 방법	문서의 특징	추정치 계산식	특징에 추정치	분류식	분류
단순 Naïve Bayes 분류자	형태소 분석 결과 나온 모든 명사	게임 구입 모습 문의 바탕 발매 버전 ..	식 (2)	P(게임|class1)=0.02121) P(구입|class1)=0.00072) P(구입|class8)=0.00092) P(모습|class2)=0.00211) P(문의|class3)=0.00564 P(문의|class8)=0.00452 P(바탕|class2)=0.00216 .	식(1)	class 2
역문헌빈도를 이용한 Naïve Bayes 분류자	1단계 : 식 (3)을 이용하여 역문헌빈도를 계산한다. 2단계 : 역문헌빈도가 높은 단어부터 정렬한 후 높은 순위의 단어만을 특징으로 채택	인상(0.890452) 이미지(0.397963) 가격(0.216647) 바탕(0.135783) 포켓(0.132654) 게임(0.132652) 차원(0.108959) 소프트(0.093939)	식 (4)	P(인상|class1)=0.00891 P(인상|class2)=0.0127) P(인상|class8)=0.00901 P(이미지|class2)=0.0201) P(가격|class1)=0.00031 P(가격|class8)=0.00201 P(바탕|class2)=0.00123 P(포켓|class1)=0.00121	식(2)	class 2
연관 단어 지식 베이스를 기반한 Naïve Bayes 분류자	Apriori 알고리즘이용	이미지&인상 ->포켓, 게임&소프트 =>버전 가격&인상 &서비스 =>제공, 이미지&인상 ->게임	연관 단어 지식 베이스를 대상으로 식에 적용 식 (9)	P(이미지&인상=>그림 |class2)=0.01211 P(게임&소프트=>버전 |class1)= 0.096154 P(가격&인상=>서비스 |class1)= 0.100775 P(가격&인상=>서비스 |class8)=0.001211 P(이미지&인상=>게임 |class1)= 0.085603	식 (10)	class 1
최적화된 연관 단어 지식 베이스를 기반한 Naïve Bayes 분류자	Apriori 알고리즘이용	이미지&인상 ->포켓, 게임&소프트 =>버전 가격&인상 &서비스 =>제공, 이미지&인상 ->게임	최적화된 연관 단어 지식 베이스를 대상으로 식에 적용 식 (9)	P(이미지&인상=>그림 |class2)=0.12011 P(게임&소프트=>버전 |class1)= 0.1315789 P(가격&인상=>서비스 |class1)= 0.138297 P(이미지&인상=>게임 |class1)= 0.117647	식 (10)	class 1

Apriori 알고리즘을 이용한 Naïve Bayes 알고리즘은 주제어를 연관 단어의 형태로 추출한다. Naïve Bayes 분류자

는 문서의 특징을 연관 단어로 추출하므로 단어의 의미 중의성으로 인한 분류의 오류를 줄인다. 예를 들어, "이미지&인상"의 연관 단어는 그래픽 클래스인 class2로 분류되며 "가격&인상"은 게임 클래스인 class1로 분류됨으로 "인상"이라는 단어에 대한 의미 중의성이 문제점으로 제기되지 않는다. 이에 따라 웹문서는 게임 클래스인 class1로 올바르게 분류된다. 반면, "가격&인상&서비스"의 연관 단어가 class1뿐만 아니라 class8에도 나타나므로 분류에 있어서 잡음이 들어간다.

최적화된 연관 단어 지식베이스를 이용한 Naïve Bayes 알고리즘은 단어 의미의 중의성을 해결하여 웹문서를 class1로 올바르게 분류하였으며 "가격&인상&서비스"의 연관 단어가 최적화된 연관 단어 지식베이스의 class8에서는 나타나지 않으므로 이로 인한 잡음을 삭제했다.

5. 성능 평가

본 논문에서는 제안된 Apriori-Genetic 알고리즘을 이용한 베이지안 자동 문서 분류(Bayesian-OAWKB)의 성능을 평가하기 위해, Apriori 알고리즘을 이용한 베이지안 문서 분류 방법(Bayesian-AWKB), 기존의 베이지안 확률을 사용한 방법(Bayesian), 역문헌빈도를 이용하는 베이지안 문서 분류 방법(Bayesian-TFIDF)과 비교하였다. 이를 평가하기 위해서 4.1절에서 기술한 학습문서와 실험문서를 사용하였다. 분류 성능을 평가하기 위해서 각 클래스로 분류된 문서를 대상으로 <표 9>과 같은 분할표를 작성한다[5].

〈표 9〉 2x2 - 분할표

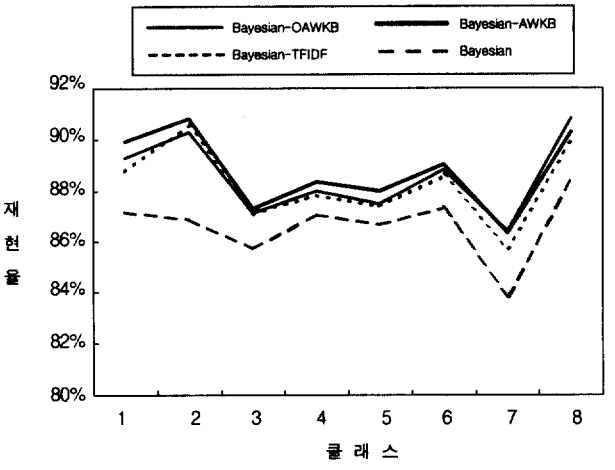
분류B / 분류A		분할 B (방법2로 생성된 분류) YES	NO
분할 A (방법1로 생성된 분류)	YES	a	b
	NO	c	d

분류의 측정은 식 (11)의 F-measure 측정식을 이용한다. 식 (11)에서 P는 정확도, R은 재현율을 의미하며, 이 경우 F-measure의 값이 클수록 분류가 우수함을 의미한다. 여기서, β는 정확도에 대한 재현율의 상대적인 가중치를 나타내는 수치로, 1.0일 경우 정확도와 재현율의 가중치가 같다.

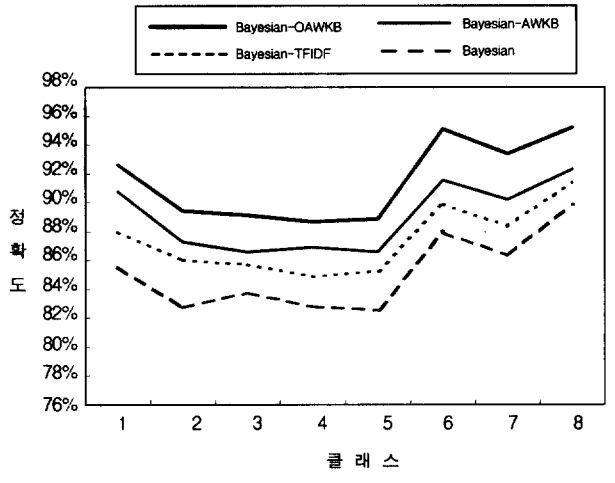
$$F\_measure=\frac{(\beta^2+1)PR}{\beta^2P+R}\quad P=\frac{a}{a+b}100\%\quad R=\frac{a}{a+c}100\% \quad (11)$$

본 실험에서는 β의 값을 1.0로 설정하여 분류 결과를 분석하였으며, 또한 β의 값을 0.5에서 1.4로 변화시키면서 F-measure의 결과 차이를 살펴보았다. (그림 7), (그림 8), (그림 9), (그림 10)은 정확도와 재현율을 식 (11)에 대입하여 분석한 결과를 나타낸다.

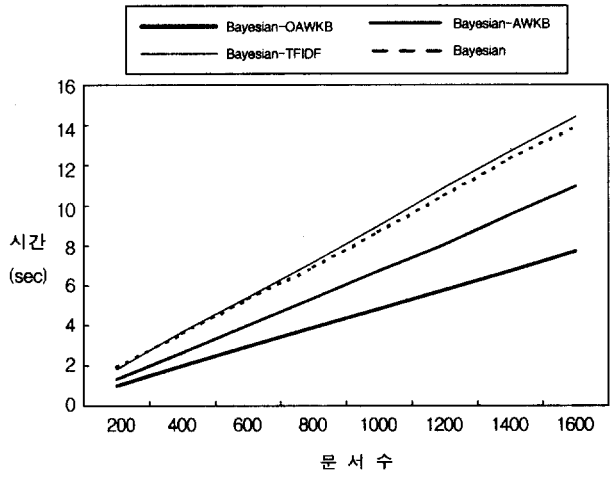
(그림 7)은 재현율을 나타내는 곡선으로 Bayesian-OAWKB 방법의 재현율이 88.53%로 Bayesian-AWKB 방법보다 0.22% 낮은 성능을 나타내며, Bayesian-TFIDF방법보다 0.30%, Bayesian방법보다는 1.90% 높은 성능을 나타낸다.



(그림 7) Bayesian-OAWKB, Bayesian-AWKB, Bayesian-TFIDF, Bayesian 문서 분류 재현율



(그림 8) Bayesian-OAWKB, Bayesian-AWKB, Bayesian-TFIDF, Bayesian 문서 분류 정확도

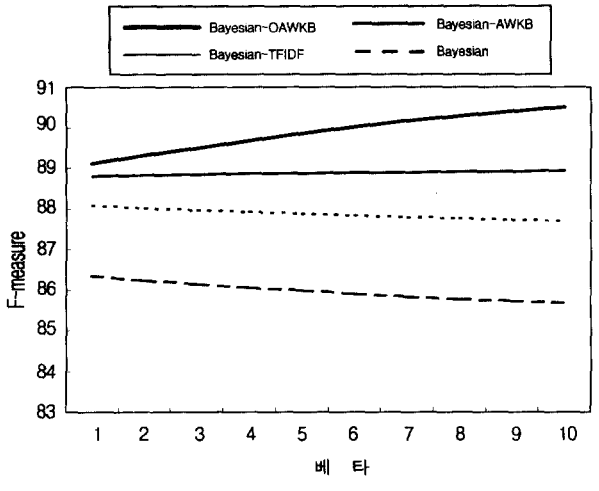


(그림 9) Bayesian-OAWKB, Bayesian-AWKB,Bayesian-TFIDF, Bayesian 문서 분류 속도

(그림 8)은 정확도를 나타내는 곡선으로 Bayesian-OAWKB 방법의 정확도는 91.55로 Bayesian-AWKB 방법보다는 2.53%, Bayesian-TFIDF방법보다는 4.12%, Bayesian 방법보다는 6.36% 높음을 나타낸다.

(그림 9)는 1600개의 문서를 분류하기 위한 속도를 나타낸다. Bayesian-OAWKB 방법은 7.72sec로 가장 빠르며, Bayesian-AWKB는 10.95sec, Bayesian-TFIDF는 14.4sec, Bayesian은 13.9sec이다. 속도면에서, Bayesian-OAWKB와 Bayesian-AWKB 방법이 다른 방법보다 우수하며, 나머지 방법들은 비슷한 속도를 나타낸다.

(그림 10)은 β값을 0.5에서 1.4로 변화시킴에 따른 F-measure의 성능 분석을 나타낸다. Bayesian-OAWKB는 β 값이 커짐에 따라 상승 곡선을 나타내므로 재현율보다는 정확도면에 높은 성능을 나타낸다. 반면, Bayesian-AWKB 방법 뿐 아니라 Bayesian-TFIDF 방법도 β값이 변할지라도 F-measure의 값은 일정한 값을 유지하므로 재현율과 정확도의 면에서 비슷한 성능을 나타낸다. 그러나 Bayesian 방법은 정확도보다는 재현율에서 다소 높은 성능을 나타낸다. 평균적으로, β=1.0일 경우, Bayesian-OAWKB 방법은 Bayesian-AWKB 방법은 1.13%, Bayesian-TFIDF 방법보다 2.18%, Bayesian 방법보다는 4.11% 높은 성능 차이를 보였다.



(그림 10) β 의 변화에 따른 F-measure에 의한 클래스별 문서 분류 성능 평가

전체적으로 연관 단어 지식베이스를 이용한 방법이 기존의 문서 내의 모든 단어의 확률을 고려하는 단순 베이지안 분류 방법과 역문헌빈도를 이용한 방법보다는 성능이 우수함을 알 수 있다. 특히, Apriori-Genetic 알고리즘을 이용한 베이지안 분류 방법이 가장 성능이 우수함을 나타냈다.

4. 결 론

본 논문에서는 기존의 단순 베이지안 분류 방법, 역문헌빈도를 이용한 베이지안 문서 분류 방법, Apriori 알고리즘을 이용한 베이지안 문서 분류 방법의 단점을 해결하기 위해, Apriori-Genetic 알고리즘을 이용한 베이지안 문서 분류 방법을 제안하였다. 제안된 베이지안 문서 분류 방법은 실험문서를 분류하기 전에 학습문서를 대상으로 최적화된 연관 단어 지식베이스를 구축한다. 그 다음으로, Naïve Bayes 분류자는 구축된 연관 단어 지식베이스를 이용하여 실험문서를 클래스로 분류한다.

본 논문에서 제안한 방법은 두 가지의 장점을 갖는다. 첫째는 Naïve Bayes 분류자가 정확할 뿐만 아니라 빠른 문서 분류가 가능하도록 최적화된 연관 단어 지식베이스를 구축했다는 점이다. 둘째는 실험문서의 특징을 연관 단어의 형태로 표현함으로써 단어 의미 중의성이라는 문제를 해결한 점이다. 본 논문에서는 제안된 Apriori-Genetic 알고리즘을 이용한 베이지안 문서 분류 방법의 성능을 평가하기 위해, 기존의 단순 베이지안 분류 방법, 역문헌빈도를 이용한 베이지안 분류 방법, Apriori 알고리즘을 이용한 베이지안 문서 분류 방법과 비교하였다. 그 결과, Apriori-Genetic 알고리즘을 이용한 방법이 Apriori 알고리즘을 이용한 베이지안 문서 분류 방법보다는 1.13%, 역문헌빈도를 이용한 베이지안 분류 방법보다는 2.18%, 단순 베이지안 방법보다는 4.11% 높은 성능 차이를 보였다.

향후, 더욱 많은 학습문서를 사용하여 최적화된 연관 단어 지식베이스를 구축하여 베이지안 분류에 이용한다면 문서 분류의 성능이 보다 높아질 것이다. 또한 문서의 특징이 단일 명사로 이루어진 연관 단어의 형태가 아닌 복합 명사로 이루어진 연관 단어의 형태로 추출된다면 문서 분류의 정확도는 더욱 향상될 것이다.

참 고 문 헌

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

[2] R. Agrawal and T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," In Proceedings of the 1993 ACM SIGMOD Conference, Washington DC, USA, 1993.

[3] H. Chen, Y. Chung, M. Ramsey, C. Yang, P. Ma, J. Yen, "Intelligent Spider for Internet Searching," Proceedings of the 30th Annual Hawaii International Conference on System Sciences-Volume IV, pp.178-188, 1997.

[4] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," ICML-97, 1997.

[5] V. Hatzivassiloglou and K. McKeown, "Towards the automatic identification of adjectival scales : Clustering adjectives according to meaning," Proceedings of the 31st Annual Meeting of the ACL, pp.172-182, 1993.

[6] Introduction to Rainbow URL : http : //www.cs.cmu.edu /afs/cs/project/theoll/www/naive-bayes.html.[7] D. D. Lewis, "Naive (Bayes) at forty : The Independence Assumption in Information Retrieval," In European Conference on Machine Learning, 1998.

[8] Y. H. Li and A. K. Jain, "Classification of Text Documents," The Computer Journal, Vol.41, No.8, 1998.

[9] M. E. Maron, "Automatic indexing : An experimental inquiry," Journal of the Association for Computing Machinery, 8 : 404-417, 1961.

[10] T. Michael, *Maching Learning*, McGraw-Hill, pp.154-200, 1997.

[11] A. McCallum and K. Nigram, "A Comparison of Event Models for Naive Bayes Text Classification," AAAI-98 Workshop on Learning for Text Categorization, 1998.

[12] J. McMahon and F. Smith, "Improving statistical language model performance with automatically generated word hierarchies," Computational Linguistics, Vol.22, No.2, 1995.

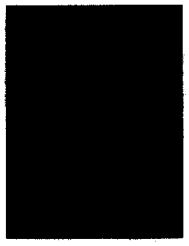
[13] 인하대학교, 사용자 중심의 지능형 정보 검색 시스템, 최종 연구 개발 보고서, 정보통신부, 1997.

[14] 정영미, 정보검색론, 구미무역㈜ 출판부, 1993.

[15] 조광제, 김준태, "역 카테고리 빈도에 의한 계층적 분류체제에서의 문서의 자동 문서 분류 시스템", 정보과학회 봄 학술발표 논문집, 4권 2호, pp.508-510, 1997.

[16] 한광록, 선복근, 한상태, 임기옥, "인터넷 문서 자동 분류 시스템 개발에 관한 연구", 제9회 한국정보처리학회 논문집, 제7권 제9호, pp.2867-2875, 2000.

[17] 허준회, 가중치가 부여된 베이지안 분류자와 단어 군집을 이용한 한국어 문서 자동 분류, 인하대학교 대학원 컴퓨터공학과 석사학위 논문, 2000.



고 수 정

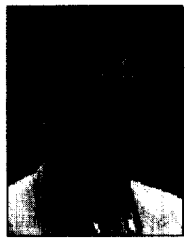
e-mail : sujung@nlsun.inha.ac.kr

1990년 인하대학교 전자계산학과 졸업 (학사)

1997년 인하대학교 교육대학원 전자계산 교육(교육학석사)

2000년 인하대학교 대학원 박사과정 수료

관심분야 : 데이터마이닝, 정보검색, 기계학습



이 정 현

e-mail : jhlee@inha.ac.kr

1977년 인하대학교 전자공학과 졸업

1980년 인하대학교 대학원 전자공학과 (공학석사)

1988년 인하대학교 대학원 전자공학과 (공학박사)

1979년~1981년 한국전자기술연구소 시스템 연구원

1984년~1989년 경기대학교 전자계산학과 교수

1989년~현재 인하대학교 전자계산공학과 교수

관심분야 : 자연언어처리, HCI, 정보검색, 음성인식, 음성합성, 계산기 구조