웹 컨텐츠의 분류를 위한 텍스트마이닝과 데이터마이닝의 통합 방법 연구

Interplay of Text Mining and Data Mining for Classifying Web Contents

최 윤 정* 박 승 수*
(YunJeong Choi) (SeungSoo Park)

요 약 최근 인터넷에는 기존의 데이터베이스 형태가 아닌 일정한 구조를 가지지 않았지만 상당한 잠재적 가치를 지니고 있는 텍스트 데이터들이 많이 생성되고 있다. 고객창구로서 활용되는 게시판이나 이메일, 검색엔진이 초기 수집한 데이터 등은 이러한 비구조적 데이터의 좋은 예이다. 이러한 텍스트 문서의 분류를 위하여 각종 텍스트마이닝 도구가 개발되고 있으나, 이들은 대개 단순한 통계적 방법에 기반하고 있기 때문에 정확성이 떨어지고 좀 더 다양한 데이터마이닝 기법을 활용할 수 있는 방법이 요구되고 있다. 그러나, 정형화된 입력 데이터를 요구하는 데이터마이닝 기법을 텍스트에 직접 적용하기에는 많은 어려움이 있다.

본 연구에서는 이러한 문제를 해결하기 위하여 전처리 과정에서 텍스트마이닝을 수행하고 정제된 중간결과를 데이터마이닝으로 처리하여 텍스트마이닝에 피드백 시켜 정확성을 높이는 방법을 제안하고 구현하여 보았다. 그리고, 그 타당성을 검증하기 위하여 유해사이트의 웹 컨텐츠를 분류해내는 작업에 적용하여 보고 그 결과를 분석하여 보았다. 분석 결과, 제안방법은 기존의 텍스트마이닝만을 적용할 때에 비하여 오류율을 현저하게 줄일 수 있었다.

주제어 텍스트마이닝, 데이터마이닝, 웹마이닝

Abstract Recently, unstructured random data such as website logs, texts and tables etc, have been flooding in the internet. Among these unstructured data there are potentially very useful data such as bulletin boards and e-mails that are used for customer services and the output from search engines. Various text mining tools have been introduced to deal with those data. But most of them lack accuracy compared to traditional data mining tools that deal with structured data. Hence, it has been sought to find a way to apply data mining techniques to these text data. In this paper, we propose a text mining system which can incooperate existing data mining methods. We use text mining as a preprocessing tool to generate formatted data to be used as input to the data mining system. The output of the data mining system is used as feedback data to the text mining to guide further categorization. This feedback cycle can enhance the performance of the text mining in terms of accuracy. We apply this method to categorize web sites containing adult contents as well as illegal contents. The result shows improvements in categorization performance for previously ambiguous data.

Keywords Text Mining, Data Mining, Web Mining

* 이화여자대학교 컴퓨터 학과
Dept. of Computer Science & Engineering, Ewha Womans University
연구분야 : 소프트웨어 및 응용/인공지능
주 소 : 127-750 서울시 서대문구 대현동 이화여자대학교
과학기술대학원 컴퓨터학과 인공지능연구실 공학관329호
전 화 : 02-3277-3505,
F A X : 02-3277-2306
E-mail : cris@ewha.ac.kr

1. 서론

최근, 인터넷이 활성화되면서 데이터베이스 기반이 아닌 무작위 형태의 새로운 데이터가 생성되는 경우가 많아지는 추세이다. 특히 전자상거래 관련 대부분의 웹 사이트에는 사용자들로부터 정형화되지는 않았지만 상당히 잠재적 가치를 지니고 있는 텍스트 데이터들이 엄청난 규모로 생성되고 있다. 이러한 환경 속에서 의사결정에 수렴할 만한 가치있고 유용한 정보를 찾아내어 분석하는 작업의 중요성이 높아지고 있다. 최근, 기업에서 유용하고 잠재적인 정보를 발견해내기 위해 많이 사용하는 데이터마이닝 기술은 정형화된 형태의 데이터를 주대상으로 하고 있다. 그러나, 대규모의 텍스트 데이터들은 구조적인 형태로 재구성하여 분석하기가 쉽지 않고, 대부분이 자연어로 쓰여진 문장 형태이기 때문에 함축된 정보를 추출하기가 쉽지 않다. 이러한 비구조적인 텍스트 문서로부터 정보를 찾아 지식을 발견하는 것이 텍스트마이닝이다[8][9]. 그러나, 텍스트마이닝은 정형화된 데이터를 위한 일반 데이터마이닝에 비하여 정보추출 능력이나 정확성 등 많이 떨어지는 경향이 있다.

본 논문에서는 이러한 문제점을 해결하기 위하여 텍스트마이닝과 데이터마이닝을 상호 보완하는 방법을 제안하고자 한다. 제안방법의 개요는 다음과 같다. 우선, 텍스트마이닝을 이용하여 클러스터링과 특성추출의 조합적용을 수행하고 그 결과를 패턴에 의한 데이터마이닝의 입력 데이터로 활용한다. 데이터마이닝의 수행결과를 분석하여 이를 다시 텍스트마이닝의 학습단계에 반영하는 반복적인 과정을 수행하는 것이다. 즉, 텍스트 데이터를 텍스트마이닝으로 일단 처리하여 정형화된 데이터로 변환하고, 이를 데이터마이닝으로 처리하여 피드백 시킴으로써 텍스트마이닝의 분류기능의 정확성을 높여주는 것이다. 이와 같은 텍스트마이닝과 데이터마이닝의 상호보완적 활용은 본 연구에서 처음 시도되는 방법으로서 두 방법이 갖는 장점을 취하면서 단점을 상호 보완할 수 있는 솔루션이 될 것으로 기대된다.

본 논문에서는 이 방법을 특성(단어)공유로 인해 분류오류가 발생할 수 있고 그 오류가 결과에 심각한 악영향을 끼칠 수 있는 분야를 택하여 실험하여 보았다. 대상 영역은 검색회사가 찾아낸 초기 데이터에서 청소년에게 유해한 사이트를 가려내는 것이다. 실험결과, 제안방법은 단순히 텍스트마이닝을 적용한 경우에 비하여 현저하게 오류율이 감소하는 것을 확인할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서 텍스트마이닝의 특징과 분석기법 등을 살피고, 3장에서 텍스트마이닝과 데이터마이닝의 상호보완을 위한 제안한 시스템 설계와 동작 시나리오의 내용을 설명한다. 4장에서는 제안한 방법에 따른 각 시스템 내용과 구체적인 수행예를 보인 후 실험을 통해 타당성을 검증하고, 마지막으로 5장에서 결론을 기술한다.

2. 텍스트마이닝과 데이터마이닝

데이터마이닝이 구조적인 데이터를 대상으로 유용하고 잠재적인 패턴을 끌어내는 것이라고 한다면, 텍스트마이닝은 자연어로 구성된 비구조적인 텍스트 안에서 패턴 또는 관계를 추출하여 지식을 발견하는 것으로, 주로 텍스트의 자동 분류작업이나 새로운 지식을 생성하는 작업에 활용되고 있다[2][3][4]. 오늘날 우리가 사용하는 대다수의 정보는 확실히 구조가 잡히지 않은 텍스트의 형태로 존재하기 때문에 자연어로 된 텍스트문서의 자동화되고 지능적인 분석은 매우 중요하다. 데이터마이닝은 많은 기업들에서 데이터간의 관계, 패턴을 탐색하고 모형화 하여 기업의 의사결정에 적용하기 위해 적용되며, 일반적인 데이터베이스와 같은 구조화된 자료에 초점이 맞춰져 있다. 따라서 데이터마이닝 작업을 위해서는 적용될 데이터가 정확하고 표준화되어야 하며, 구조화가 잘 되어진 후에야 비로소 적용할 수 있을 것이다. 이를 위하여 최근에는 XML과 같이 텍스트를 구조화하려는 시도가 활발하게 이루어지고 있으나 아직 텍스트 문서들은 비구조적인 형태가 대부분이다.

2.1 텍스트마이닝

텍스트마이닝 기술체계는 자연어처리, 정보추출, 시각화, 데이터베이스 그리고 기계학습의 분야를 포함하고 있다. 텍스트마이닝에서 가장 일반적으로 사용하는 기법은 *특성벡터(feature vector)*를 이용하는 것이다. 이 방법은 *특성추출(feature extraction)*과정을 통하여 텍스트에 대한 특성벡터를 생성하게 된다. 따라서, 텍스트 분석의 기반이 되는 것이 바로 특성추출에 의한 특성벡터이며 이의 통계수치는 각 분석기법들의 근거가 되는 것이다.

특성추출(feature extraction)은 텍스트에서 중요한 용어(term)를 인식하여 추출해 내는 것으로, 추출된 용어들은 일반적으로 단어의 원형(word)으로 변형되어 특성벡터를 구성하게 된다. 이러한 특성벡터는 문서를 분류하거나 요약하는데 기초정보로 사용되며, *특성(feature)*의 중요성을 나타내는 가중치 함수와 지지도함수의 계산은 단어가 발생한 위치와 발생한 횟수에 기반한다. 가령, 한 문서 내에서 여러 번 나타나는 단어의 중요도는 높다고 가정하지만, 여러 문서에서 걸쳐서 나타나는 발

나는 발생도가 높다면 이 단어의 중요도는 낮다고 간주한다. 따라서 가중치 부여함수의 계산에는 단어가 발생한 문서 개수의 역함수(inverse)값이 사용된다[14][15]. 이에 기반한 기법은 텍스트에서 정보나 지식을 발견하고 추출하는데 사용되며 그 방법에 따라 크게 문서의 군집화, 분류화, 요약의 세 가지로 분류된다. 여기에서 중요한 것은 이들 세 가지가 모두 단순한 특성벡터에 기반하고 있다는 사실이다[1][2][3][8][9][11].

2.2 데이터마이닝

데이터마이닝은 데이터베이스의 데이터처럼 정형화된 데이터를 대상으로 처리하기 때문에 텍스트마이닝에 비하여 특성간의 연관성 파악이나 규칙생성 등 매우 다양하고 강력한 알고리즘들이 많이 개발되고 있다. 특히 분류작업의 경우 특성벡터에 의존하는 텍스트마이닝에 비하여 결정트리, 신경망, 연관규칙 등 다양한 알고리즘이 지원될 수 있다. 본 연구에서는 이러한 기능을 기존의 텍스트마이닝에 적용하기 위한 방법을 제안하는 것이다.

2.3 마이닝 도구

일반적으로 데이터마이닝과 텍스트마이닝 도구는 사용자가 시나리오를 작성하고 데이터준비와 분석을 도울 수 있도록 상용 혹은 비상용으로 많이 개발되어 있다[20][21][22][23].

본 논문에서는 텍스트마이닝과 데이터마이닝의 혼합 설계를 위하여 텍스트마이닝 도구로는 IBM사의 "Intelligent Miner for Text"(이하 IMT)[20], 데이터마이닝 도구로는 SPSS사의 "Clementine"을 사용하였다[23].

1) 텍스트마이닝 도구 : IMT

본 연구에서 텍스트 분석을 위해 사용한 텍스트마이닝 도구는 IMT이다. IMT에서 제공하는 분석 기능으로는 *언어식별(language identification)*, *주제분류(topic categorization)*, 국가별 사전 데이터를 이용하여 중요한 용어나 이름, 약어, 장소 등을 나타내는 어구들을 자동으로 인식하는 *특성추출(feature extraction)*, 유사한 문서집합을 자동으로 그룹이나 클러스터로 나눌 수 있는 *군집(clustering)*, 문장을 분석하여 문서의 요약정보를 추출하는 *요약(summarization)*등이 있다[8][20]. IMT는 현재 상용화된 대표적인 텍스트마이닝 도구이긴 하지만 단순히 특성벡터 기법에 기반하고 있기 때문에 정확도가 떨어지는 단점이 있다. 일반적인 텍스트 분석결과가 항상 최상위 순위로 결정되기 마련인데 이는 간단하고 빠르긴 하지만 정확도와 신뢰도가 낮아지는 주된 원인이 된다. IMT의 분석도구는 Unix나 DOS의 명령어 형식으로 제공되어 분석가나 일반인들이 텍스트 분석을 위해 사용하는 데는 어려움이 따르나 이러한 특징이 분석기법의 순차 혹은 조합적인 적용으로 주어진 문제해결을 위한 응용프로그램을 만들어 활용하는 데는 더 유리하다. IMT의 이러한 특징들은 기본 기능을 조합하고 응용하여 문서의 범용적인 분석이나 특수한 목적을 지닌 분석작업에 있어서 복잡한 처리사항을 고려하여 반복적인 분석작업을 가능하게 하고 있다.

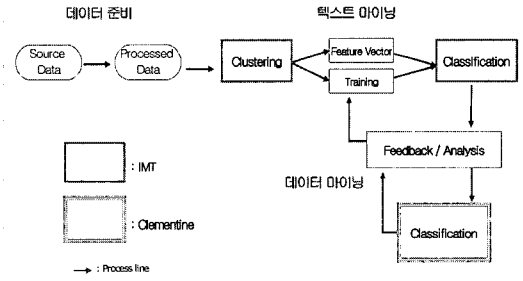
2) 데이터마이닝 도구 : Clementine

본 연구에서 데이터 분석을 위해 사용한 데이터마이닝 도구는 Clememtine이다. SPSS에서 개발한 Clementine에서 제공하는 방법론은 일반 데이터마이닝에서 수행하는 군집, 분류, 연관, 시각화 등을 포함하고 있다. 특히, 작업공간과 단계별 작업들이 순서에 맞게 나열되어 있어서 분석의 전 과정을 제어하고 시각적으로 이해하기 쉬우며 또한 설명하기 수월하다는 장점이 있다. 또한, 데이터 분류를 위하여 결정트리, 신경망, 연관규칙 등 다양한 알고리즘이 제공된다. 단, 데이터마이닝의 경우 입력형태가 데이터베이스의 데이터처럼 정형화 되어있거나 전처리 과정을 통하여 쉽게 정형화시킬 수 있는 데이터에 국한된다.

3. 웹 컨텐츠 분류를 위한 마이닝시스템 설계

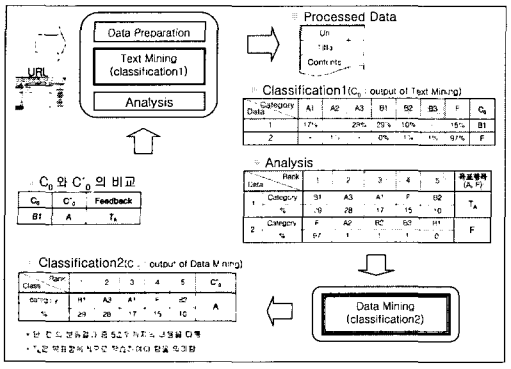
3.1 전체시스템 구조

(그림 1)은 본 연구에서 제안한 전체 시스템을 설명하는 구성도로서 각 시스템이 동작하는 내용을 설명하고 있으며, 각 단계는 '데이터 준비' - '텍스트마이닝 분류' - '결과분석' - '데이터마이닝 분류' - '피드백'의 순환과정으로 정리된다.



(그림 1) 전체 시스템 구성도

(그림 2)는 데이터가 입출력되는 과정으로 각 과정의 출력결과는 다음과정의 입력 데이터로 사용되며 원하는 결과를 얻을 때까지 순환하는 동작을 나타낸다.



(그림 2) 데이터 입출력

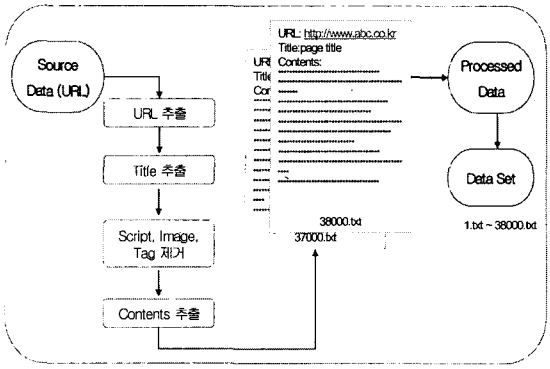
텍스트마이닝 분류의 입력은 문서들이며 분류결과는 각 문서에 대하여 지정될 수 있는 항목(category)들과 신뢰수치(score)로 나타난다. 데이터마이닝 분류의 입력값은 텍스트마이닝 결과를 분석하여 얻어낸 의미를 내포한 패턴들이며 분류결과는 각 문서에 대하여 지정될 수 있는 항목으로 나타난다. 단, C_0와 C'_0 는 4.4.3의 (그림 18)에서 설명하고 있다.

우선, 텍스트 분석이 가능하도록 분석에 사용되어질 정보를 추출해 내고 불필요한 것은 제거하는 텍스트정제 과정에서 시작하여 적절한 텍스트 분석기법 적용을 통해 분류를 수행하는 텍스트마이닝 과정으로 진행한다.

3.2 단계별 상세 구조

3.2.1 데이터 준비

효과적인 전처리를 위해서는 능률적으로 부가적인 단어와 형태소를 여과하고 불필요한 요소를 제거하기 위해 사전을 통한 미세한 조정을 필요로 한다. 이상적인 전처리 단계에서는 어떠한 의미도 전달하지 않는 조사와 같은 보충적인 단어들을 없애고, 접두사와 접미사, 형태소를 분리시키면서 단어의 어간(stem)을 확인하는 작업이 수행된다[25]. 이러한 정제과정을 통해 분석과정에 적합한 최적의 데이터 상태를 만들어 분석의 질을 향상시킬 수 있는데, 전처리 작업은 실제 분석에 소요되는 시간보다 더 많이 걸릴 수 있으며 수집한 데이터를 잘 이해하는 일이 필요하다. 특히, 데이터 준비와 분석 단계에는 입력 데이터의 형태와 성격에 따라 각기 다른 처리가 요구된다.



(그림 3) 웹 문서의 분류분석을 위한 전처리 과정

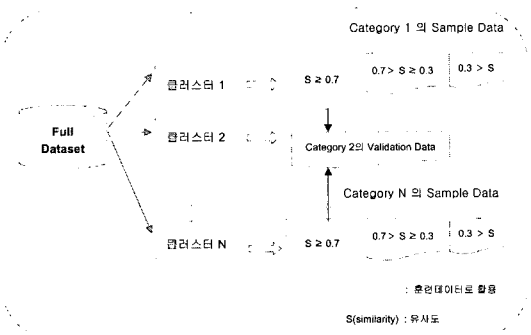
본 연구의 분석대상이 되는 데이터는 모두 웹 문서로서 형태의 특징을 고려하여 (그림 3)과 같이 전처리하였다. 각 URL에 해당하는 웹 페이지 수집 후, 이미지 파일과 불필요한 태그, 스크립트 등의 특정 문자들을 제거하여 정제된 파일을 생성하고, 전체 파일이름을 갖는 문서집합을 구성한다.

3.2.2 텍스트마이닝

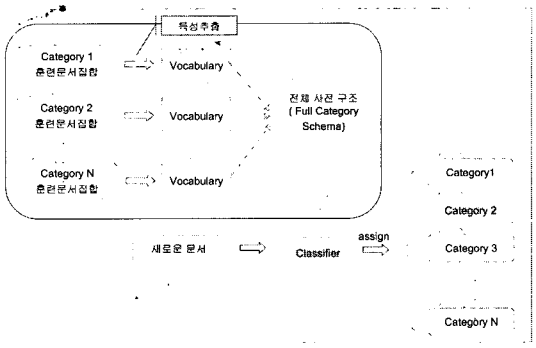
이 과정은 텍스트마이닝에 의한 1차 분류를 수행하는 단계로 비구조적인 텍스트위주 문서에 다양한 텍스트 분석기법을 적용함으로써 지식을 발견해 낸다. 적용하는 주요 분석기법은 *군집(clustering)*과 *분류(classification)*이며, 분류분석 수행에 앞서 군집화를 먼저 수행시켜 전체 문서집합의 개요를 획득하고 분류를 위한 판단기준을 얻어낸다. 즉, 군집은 분류의 준비단계로서 사용자가 i)분류해 낼 항목(category)을 명확히 정의하고 ii)각 항목에 따른 훈련문서를 선정하여 학습시키는 과정에 군집결과를 이용하는 것이다.

1) 군집(clustering)을 이용한 개요파악 및 샘플링

군집분석은 데이터에 대한 기반지식 없이 분석 초기에 행하여 결과를 분석할 수 있다는 장점이 있으며, i)중복 혹은 유사한 문서를 제거하고 ii)다른 문서의 주제와 다른 주제를 가진 문서를 구별하고 iii)대량의 문서집합의 개요를 획득하는 데 적용할 수 있다. (그림 4)는 군집을 수행하여 생성된 각 클러스터들을 나타내며, 의미상 분류된 클러스터들에 의해 ii)와 iii)의 응용이 가능함을 의미한다. 각 클러스터에 특성추출을 행한 결과로 나타나는 단어들에서 전체 데이터가 포함하는 주제나 성격이 무엇인지 감지할 수 있으며 실제 분류해 내어야 할 기준을 얻을 수 있다.



(그림 4) 군집을 이용한 데이터 개요 획득 및 데이터 샘플링



(그림 5) 분류도구에서 사전구성을 위한 학습과정

군집을 통해 생성된 각 클러스터 내에는 공통 특징을 공유하여 서로 높은 유사도를 가진 문서들도 존재하기 마련이다. 특정 임계값 이상의 유사도를 갖는 문서들은 그 클러스터의 성격을 잘 나타낸다고 볼 수 있으며, 다른 클러스터와 구별되는 특성이 되기 때문에 분류작업의 준비단계인 샘플링(sampling) 과정에 각 클러스터 내의 문서들을 활용한다. 즉, 학습 문서를 선정하는 데 있어서 주제를 내포하는 문서를 개별적으로 선택하기에 앞서 서로 긴밀하게 뭉쳐진 문서들 중에서 우선적으로 선정하는 것이다.

2) 분류를 위한 학습(training)

군집(clustering)과 달리 분류(classification)를 수행하기 위해서는 각 항목을 위한 학습데이터를 사용자가 선정하여 훈련시키는 과정이 필요하다. (그림 5)는 위에서 선정된 각 훈련문서에서 특성을 추출해 내어 *특성벡터(feature schema)*를 구성하는 기본 학습과정을 보이고 있다[8]. 특성벡터에는 추출된 각 특성의 성격 및 발생빈도와 발생위치에 따른 값과 가중치, 그리고 그 외의 부수적인 값들이 함께 부여되어 분류의 근거가 된다. 따라서 *학습(training)* 과정이 진행될수록 특성벡터의 크기도 증가하게 되는데, 이는 *분류기(classifier)*의 지식이 점차적으로 확장되어 감을 의미하는 것이다. 결과적으로, 분류기는 새로운 문서에 대해 어떠한 항목(category)으로 지정할 것인지에 대한 정보를 *특성벡터*를 통하여 얻음으로써, 자동으로 분류를 수행하게 된다.

3) 분류(classification)

일반적인 텍스트마이닝 분류 결과는 각 문서에 대해 분류항목으로 정의된 항목(category)들과 각 항목에 대한 신뢰도의 점수치(score)로 나타난다. 이 점수치는 각 항목에 대한 상대적 혹은 절대적 수치 값으로 나타낼 수 있는데, 대부분의 경우 분류지정 방식에서 가장 높은 신뢰도를 갖는 1순위 항목으로 지정하거나 단순히 수치에 의존한 방법을 이용한다. 이를테면, 절대적인 신뢰수치가 현저하게 낮은 문서들은 내용의 가치가 떨어진다는 의미를 가짐으로써 우선적으로 가려낼 수 있다. 그러나 이러한 방법으로 텍스트마이닝에 의한 분류결과를 개선하는 데에는 한계가 따른다. 각 항목의 수치가 같을 경우에는 하나이상의 항목에 지정될 수 있고, 항목간 근소한 신뢰도 값의 차이를 보일 경우 최상위 항목에 지정되므로 다른 분석과정 없이 단순판단에 의존한다면 정확도가 낮아지고 에러율이 높아지게 되는 문제점이 있다. 그러므로, 분류 결정방법에 있어 앞서 제기한 문제점을 해결하기 위해 기존의 수치에 근거한 방법과 더불어 항목의 의미를 고려한 방법을 제안한다.

3.2.3 분류결과의 분석 및 피드백

텍스트마이닝 분류결과를 분석하기 위해 우선 다음과 같이 몇 가지 사항을 정의한다.

- *정의 1* : 후보항목(candidate category)은 대상문서에 대하여 텍스트마이닝 결과로 나타나는 항목들의 리스트를 말한다.
- *정의 2* : 목표항목(target category)은 궁극적 분류 대상이 되는 항목들을 말한다.
  (본 논문의 예에서 A, F항목이 이에 해당한다.)
- *정의 3* : 미정항목(intermediate category)은 현재 사용되는 항목 중에서 목표항목이 아닌 것을 말한다. (본 논문의 예에서 Bi항목과 T항목이 이에 속한다.)

<표 1>은 일반적으로 나타내는 텍스트마이닝의 분류 결과의 각 후보항목(candidate)들을 정규화하여 데이터화한 예이다. 여기에서 'data_id=634227'의 경우 추정 분류값이 1순위가 F항목으로 97%의 신뢰도를 갖고 2순위가 A2로 1%의 신뢰도를 갖는다는 뜻이다. 이러한 추정치를 5순위까지 나타내고 있다. 여기에 나타낸 신뢰도의 수치는 IMT시스템에서 생성한 점수치(score)를 100으로 정규화시킨 값으로서 이를 rank score라고 부른다. total score는 정규화 시키기 이전의 각 분류항목 수치의 합을 나타내는 것으로써, 전체 total score들의 평균치보다 현저히 낮은 값을 가지는 문서의 경우에는 결과를 신뢰할 수 없다고 가정한다. 그러므로 각 total score는 데이터별 신뢰치를 나타내고 rank score는 한 데이터에 대한 분류항목간의 상대적 신뢰치를 나타낸다.

<표 1> 텍스트마이닝 분류결과의 후보항목리스트 정규화 (Total Score의 평균값: 442.07)

Rank Score / Data id	1	%	2	%	3	%	4	%	5	%	Total Score	Assign	분석 결과
634227	F	97	A2	1	B2	1	B3	1	B1	0	726.33	F	유효
1114467	F	74	B3	11	A3	6	A1	5	B2	3	662.8	F	유효
1678389	B1	29	A3	28	A1	17	F	15	B2	10	514.42	B1	무효
10389	A3	74	F	9	B1	7	A2	6	A1	4	139	A3	무효

기존 텍스트문서 분류를 위한 시스템의 경우 후보항목의 rank score를 고려하지 않고 일률적으로 최상위 항목으로 결정하기 때문에 항목간의 rank score의 차이가 근소할 때와 클 때의 구분이 없어 정확도가 매우 낮다. <표 1>의 'data_id=1678389'의 경우 1순위 항목인 B1으로 결정되었으나 1, 2순위간 편차가 1%에 불과하므로 신뢰할 수 없는 결과라고 볼 수 있다. 이러한 데이터는 서로 같은 특성을 일부분 공유하고 있지만 분명히 다른 카테고리에 해당하는 문서들에 해당한다. 또한, 'data_id=10389'의 경우 1순위 rank score가 74%인 높은 신뢰도를 갖지만 정규화되기 이전의 모든 rank score의 합인 total score가 전체 total score의 평균값 442.7보다도 매우 낮은 값인 139로서, 이 역시 결과를 신뢰할 수 없다. 이러한 데이터는 내용이 빈약하여 품질이 낮은 문서에 해당한다. 이와 같은 데이터에 대해 모호성이 강하다고 간주하고 따로 선별하여 고려할 수 있도록 구분하여 놓는다.

본 논문에서는 이와 같은 문서들을 분류하기 위한 방법으로 다음과 같이 제안한다. 1차 결과에서 해결하지 못한 데이터에 대해 rank score와 항목의 순위차(distance)가 반영된 사례를 주어 교사학습 시킨 후 2차 분류(데이터마이닝 분류)를 수행하여 결과를 얻어낸다. 이 때, 1차분류와 2차분류의 결과를 비교하여 1차 분류의 학습 과정에 반영한다. 이 과정은 '1차분류(텍스트마이닝)결과 분석 - 2차분류(데이터마이닝) - 해석 및 재학습 - 1차분류 수행'으로 진행된다. 이 때, 1차분류에 대한 분석과정은 1)Rank Score와 Total Score를 고려 2)항목의 순위차(distance) 고려 3)Rank Score와 순위차를 모두 고려하는 3단계로 구성된다. 이 중 1단계는 텍스트마이닝의 결과를 이용한 것이고, 각 단계에서 고려된 결과는 데이터마이닝의 입력값으로 주어질 학습 패턴으로 주어진다. 그리고, 통합시스템을 이용하여 이 과정을 반복 수행하게 된다. 이를 자세히 설명하면 다음과 같다.

■ Step 1 : Rank Score와 Total Score를 고려

위의 <표 1>에서 설명한 각 항목간의 신뢰도 격차를 이용하여 문서 분류의 기준을 삼는 과정이다. 이 과정에서 일정기준의 total score를 만족하는 문서에 대해서만 분류결과를 인정한다. 이는 연관규칙 알고리즘에서의 지지도(support)의 의미에 해당하며 사용자가 파라미터로 제공한다. 일단 기준을 만족하면 순위별로 주어진 rank score를 이용하여 1위와 2위의 격차가 일정기준 이상을 만족하는 경우 1순위 항목으로 지정한다. 이는 연관규칙에서 신뢰도(confidence)의 의미에 해당한다고 볼 수 있다. 이를 위하여 다음과 같은 형태의 규칙을 사용한다.

in Document D

If (D.total_score ≥ Min_Support) then

If (D(1).rank_score ≥ Max_Value) then Assign D to Top_Rank_Category

If ((D(1).rank_score - D(2).rank_score) ≥ Min_Value) then Assign D to Top_Rank_Category

Else goto Step 2

위에서 min_support, max_value, min_value는 모두 사용자가 제공하는 파라미터이다.

위의 규칙은 total score가 일정 기준(min_support)이상인 문서에 대해, 1순위 항목의 rank score가 일정수치(max_value)를 초과하거나 1, 2순위의 격차가 일정수치(min_value) 이상일 경우 1순위 항목으로 인정하고, 그 외의 경우에는 다음단계에 의한 분석을 수행하라는 의

미이다. 이러한 각 일정 수치들을 임의로 정할 수 있으나 4장의 <표 2>에서 나타낸 rank score의 분포도와 격차에 따라 조정하는 것이 바람직하다.

Step 1에서는 적정값 이상의 total score를 받은 문서들에 대하여 충분히 변별력 있는 rank score가 주어진 문서들에 대해서만 항목을 지정하여 주었다. 이 때 Step 1에서 항목이 지정되지 못한 문서를 미지정 문서라 부르기로 한다. Step 2에서는 이러한 미지정 문서를 대상으로 후보항목 패턴을 감안하여 항목을 지정하도록 한다.

· *정의 4* : 피보트항목(pivot category)은 미지정 문서의 후보항목 중 순위가 가장 높은 미정항목을 말한다. 단, 피보트항목에 순위적으로 인접한 다른 미정항목들은 피보트항목에 병합시켜 같은 항목으로 간주한다.

■ Step 2 : 항목의 순위차(distance) 고려

Step 1에서 미분류된 문서를 위해 항목간의 순위차를 고려하는 과정이다. 이 경우 미지정 문서는 항목을 부여할 근거가 충분치 못하기 때문에 순위가 가장 높은 미정항목을 중심으로 목표항목과의 거리를 계산하여 항목을 지정한다. 거리계산을 위하여 항목 x, y사이에 거리함수 Dist(x,y)를 다음과 같이 정의하여 사용한다. 단, RD(x,y)는 x와 y의 순위차를 의미한다.

$$Dist(x, y) = RD(x, y) \times weight \quad (1)$$

$$weight = \log(\sqrt{rank + 0.1}) \quad (2)$$

예를 들어 x와 y의 항목이 인접해 있으면, RD(x,y)는 1이고, 그 사이 다른 항목 z가 있으면 2가 된다. 이 때, 순위별로 중요도를 감안하여 다음과 같이 가중치를 정하여 사용한다. 여기에서 rank는 항목의 절대 순위값이며, 가중치를 주는 이유는 순위간의 거리에 따라 현저한 격차를 부여하기 위함이다. 식 (2)에서 순위에 0.1을 더한 것은 단순히 기술적인 이유 때문이며, 상황에 따라 조정될 수 있다. 거리를 근거로 한 항목의 지정은 다음과 같이 진행된다.

$$\min_{n} \{Dist(C, Cn)\} \quad (3)$$

식 (3)을 만족하는 Cn항목으로 미지정 문서의 항목을 지정한다. 여기에서 C는 피보트항목이고 Cn은 목표항목이다.

■ Step 3 : Step 1과 Step 2의 결과를 이용하여 훈련 데이터 생성

Step 1과 Step 2의 과정에서 살펴본 사항들을 모두 고려하여 데이터마이닝 분류의 학습을 위한 데이터를 생성한다. 이의 구체적인 입력형태는 4장의 (표 4)와 같다. 이 입력을 이용하여 데이터마이닝을 수행하면 후보항목의 패턴을 조건항으로 갖는 규칙이 생성된다. 이 규칙은 피드백되어 텍스트마이닝 분류의 근거로 사용된다.

4. 시스템의 구현 및 적용

본 장에서는 3장에서 제안한 마이닝 모델로서 문서 데이터의 분류를 위해 설계한 (그림 1)의 각 시스템을 구현하고 실제 데이터의 분석과정과 실험결과를 보인다.

실험 대상은 검색회사가 찾아낸 초기 데이터이며, 유해사이트를 가려내어 분류 품질의 향상을 기하려는 것이다. 구현한 내용은 각 URL에 해당하는 웹 문서의 수집과 정제를 위한 *전처리기*, 피드백을 통한 재학습-재분류의 반복 작업이 가능한 *텍스트마이닝 분류분석기*, 그리고 데이터마이닝 분석을 위해 결과를 데이터화하는 *변환기*이다. 이 응용프로그램들은 Windows NT상에서 IMT와 VC++를 사용하여 구현하였으며, 본 장에 기술된 실험내용은 위의 응용프로그램으로 작성한 마이닝 솔루션으로 수행된다.

4.1 시스템 동작 시나리오

다음은 실험을 위한 세부시나리오 내용이다.

□ 데이터 준비

단계 1:수집	url에 해당하는 온라인문서 수집
단계 2:정제	불필요한 코드가 제거된 url/title/contents 형식의 문서로 재저장
단계 3:문서집합생성	full dataset 파일생성

□ 텍스트마이닝 분류 수행

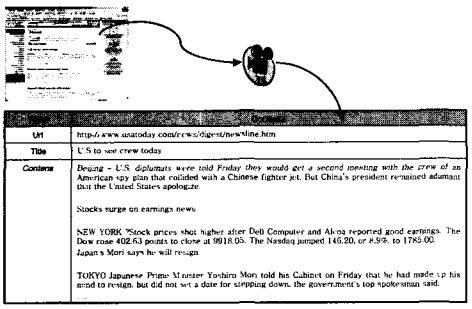
단계 1: 군집분석	대량의 문서집합에 대한 개요 획득
단계 2: 특성추출	이 름 / 용 어 / 관 계 등 을 추출하고, 분류수행시 준비작업에 필요한 정보획득
단계 3: 학습	단계1,2를 통한 항목 선정 및 학습 수행
단계 4: 분류분석	항목과 항목의 신뢰도 수치로서 나타나는 분류결과 획득

□ 결과 분석 및 피드백

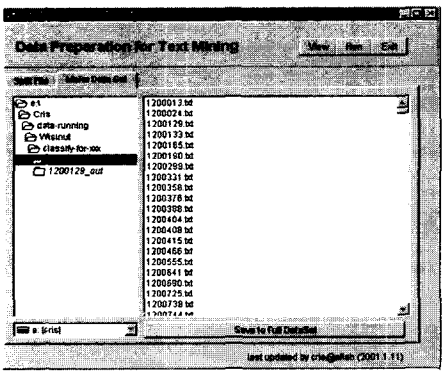
단계 1 : 정규화	분류결과를 %로 정규화시킨 후 반구조화일로 변환
단계 2 : 분류결과 분석	Rank Score와 항목의 의미를 고려하여 패턴별 판단사례를 구축한 후, 2차 분류분석 수행
단계 3 : 해석	1,2차 분류결과의 비교-해석 후, 훈련 데이터의 재선택/통합/여과의 반복수행으로 놓쳐버릴 수 있는 특징 추출
단계 4 : 피드백	텍스트마이닝 분석의 학습단계에 반영-재학습/재분류

4.2 데이터 준비

실험에 사용한 데이터의 대상분야는 검색회사가 찾아낸 약 38000건의 URL집합이며, 4.1의 데이터 준비과정 시나리오를 진행한다. (그림 6)은 웹 문서를 전처리한 화면이며, (그림 7)은 전처리된 각 문서들을 텍스트마이닝 분류를 위한 입력으로 사용하기 위해 전체 문서집합으로 형성하는 화면이다.



(그림 6) 데이터 전처리



(그림 7) Input Stream 을 위한 Dataset 작성

4.3 텍스트마이닝 분류수행

1) 군집(clustering)

군집분석의 수행결과로 전체 문서집합들은 약 20개의 클러스터를 이루었으며, (그림 8, 9)는 각각 1번과 9번 클러스터에 해당하는 문서들의 특성을 보이고 있다. 1번 클러스터의 문서들에서 추출된 단어들은 'make, people, take, time, will, use, say, become, like' 등의 두드러진 특징없는 평이한 단어들이지만, 9번 클러스터의 문서들에서 추출된 단어들은 'gay, hot, teen, free, video, hardcore' 등으로 나타났다. 따라서, 이에 속하는 문서들은 유해사이트일 경우가 매우 높으며 차별적으로 분류되거나 제거되어야 한다는 결론을 얻고, 이를 분류 목표와 항목을 정의하는 데에 이용한다.

Cluster 1
Contains 99 document(s)
Best descriptors for this cluster
make, people, take, time, will, use, say, become, like
Most frequent descriptors for this cluster
http, URL, will, make, time, take, use, people, year, look

(그림 8) 1번 클러스터의 특성

Cluster 9
Contains 7 document(s)
Best descriptors for this cluster
click, gay, hot, teen, free, video, hardcore, xxx
Most frequent descriptors for this cluster
http, URL, click, gay, sex, teen, site

(그림 9) 9번 클러스터의 특성

2) 분류(classification)

위 군집분석의 반복수행 결과로 인해 전체 데이터 집합에는 일반적인 뉴스 · 광고 · 오락내용을 다룬 평이한 문서와, 자살 · 폭력 · 음란성을 띄고 있어 이를 접하게 되는 청소년이나 어린이들에게 유해한 문서들이 모두 섞여 있음을 알 수 있다. 특히 '성'이나 '광고', '교육'과 관련된 단어를 공유하여 유해 가능성이 있는 문서들이 포함되어 있어 분류의 오류로 인해 악영향을 미칠 수 있다는 결론을 얻을 수 있다. 이러한 문서들이 앞서 언급한 의미상 모호한 문서에 해당되어, 미정항목으로 지정된다. 이러한 문서들은 기존 텍스트마이닝에 의한 분류 시 '교육'이란 단어를 다량 포함하고 있으므로 폭력 성교육에 관한 문서가 무해하다고 결정될 수 있으며, '성'이란 단어를 포함하고 있으므로 성교육에 관한 문서

A1	A2	A3	B1	B2	B3	F	T
경고메시지나 legal -term 를 포함	적극적으로 접속자의 신분 확인을 요구	A1,A2에 해당되지 않음	광고	교육	뉴스	일반적인 정치,경제,과학, 상품과 관련된 뉴스,교육,광고	1차 분류결과의 미정문서

•목표항목 A: 크게 유해하고 부적합한 내용
•목표항목 F: 누구에게나 적합한 내용
•미정항목 B: 어린이나 청소년처럼 일부 계층에게는 부적합한 사례
•미정항목 T: 1차결과 B로 지정된 문서

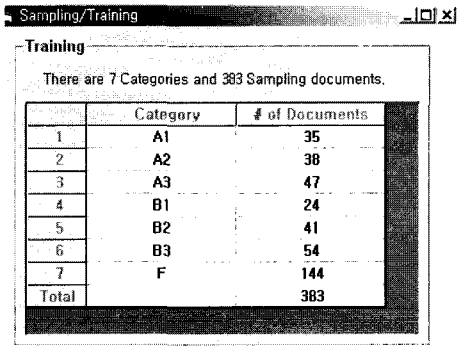
(그림 10) 분류를 위한 항목 정의

가 유해하다고 결정될 가능성이 있는 문서들이다..

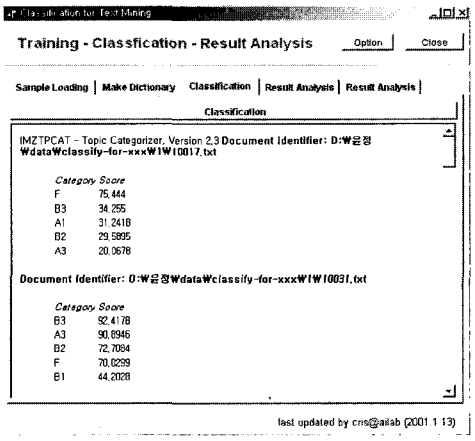
본 연구의 분류 목표는 청소년과 어린이의 관점에서 본 유해문서의 선별이며, (그림 10)과 같이 세분화된 항목을 정의하여 분류하기로 한다. 분류 항목의 내용은 다음과 같다. 매우 유해한 내용의 문서를 의미하는 A, 일반적이고 보편적인 내용의 문서를 의미하는 F, 관점에 따라 A와 B사이에서 관점에 따라 서로 상반된 결정을 내릴 수 있는 B 그리고 다음 단계에 반영되어야 할 지침사항을 알리기 위한 T의 네 그룹으로 나눈다. 이 때, 목표항목은 A와 F항목이며, B와 T항목은 미정항목에 속한다. 여기서 T항목은 학습 단계에서 목표항목으로 학습시켜야 함을 나타내어 피드백사항을 지시한다. 또한, B항목은 A 또는 F로 분류될 수 있는 모호한 문서들의 그룹을 의미하며, 이는 실제 데이터에는 텍스트마이닝 분류에 의한 단순한 1순위 지정방식에 의해 결정이 엇갈릴 수 있는 사례가 빈번하다는 근거를 감안한 것이다. 텍스트마이닝 분류결과에 대한 분석에 있어서 이러한 미정항목은 각 항목들의 순위값과 거리차를 고려한 패턴을 고려하는 데에 응용할 수 있다.

위의 각 항목마다 부가적인 세부 항목을 추가할 수 있으며, 다음의 예와 같이 학습시킨다. 유해사이트이면서 경각심을 충분히 표현하거나 내부규율을 갖는 A1, A1의 내용이면서 적극적으로 접속자의 신분 확인을 요구하면 A2, 아무런 경고 내용이 없는 불법적인 문서는 A3으로 학습시킨다. 또한, 일부 계층을 위한 특정 광고 · 쇼핑 · 교육사이트는 각각 B1과 B2로 학습시킨다.

(그림 11)는 입력값으로 각 목표항목과 훈련데이터를 주어 학습하는 과정으로 이는 결과분석 및 피드백 과정을 통해 반복 수행된다. (그림 5)의 학습과정에 따라 훈련문서를 선정하여 이로부터 특성추출을 통한 특성벡터를 구성한다. 훈련문서는 A항목 120건, F항목 144건, B항목 119건으로 총 383건의 문서를 선정하였다.

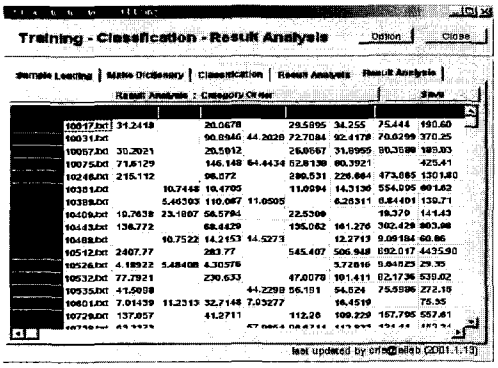


(그림 11) 훈련문서 Loading

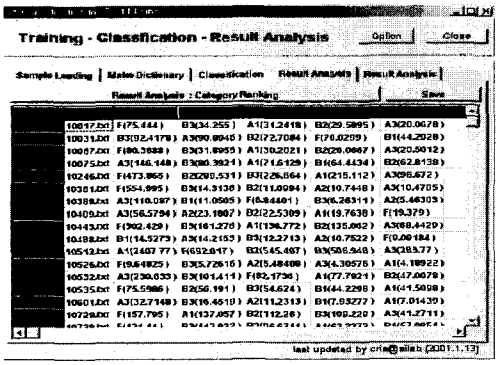


(그림 12) 텍스트마이닝 분류결과

(그림 12)는 IMT에 의한 분류수행 결과로서 일반적인 텍스트분류 결과와 같이 각 문서에 대한 *항목(category)*과 *점수치(score)*의 리스트로 나타난다. (그림 13, 14)는 각각 항목별/순위별 결과로 분석한 화면이다.



(그림 13) 항목별 분석



(그림 14) 순위별 분석

4.4 분석 및 피드백

Rank 패턴이 반영되지 않아 근소한 차이의 경우 에러율이 높아진다는 점을 개선하기 위해, 3.2.3의 '결과분석 및 피드백' 과정에서 제안한 방법으로 결과분석을 위한 시나리오를 진행한다. 텍스트마이닝의 분류결과를 데이터화하여 패턴을 고려한 판단사례를 주어 데이터마이닝 분류를 수행한다. 이는 텍스트마이닝 결과와 비교분석하여 텍스트마이닝 분류의 학습과정에 피드백되는 것이다. 즉, 새로운 학습대상을 수집하여 (그림 11, 12)의 학습 프로세스를 재실행한다.

4.4.1 분류결과에 대한 분석

텍스트마이닝 분류결과를 분석하여 데이터마이닝 분류를 위한 입력값으로 이용할 데이터를 선정하는 과정이며, 3.2.3장의 각 단계에 대한 실제 사례와 함께 설명한다. 단, 본 실험에서는 텍스트마이닝 분류의 5순위까지의 결과를 얻어 100(%)으로 정규화하고 테이블 폼으로 변환하여 분석한다.

■ Step 1 : Rank Score와 Total Score를 고려

정규화한 데이터로 <표 2>와 같이 전체 1순위가 갖는 rank score의 분포를 얻는다. 이 때 1순위 신뢰도가 70%으로 다른 분석 필요 없이 바로 1순위로 지정할 수 있는 데이터는 전체 데이터의 10%에도 미치는 못하는 181건에 불과하다. 본 연구에서는 데이터 분포도에 기반하여 최대신뢰도 50%, 최소신뢰도 20%인 경우 최상위 항목으로 지정하도록 하였다. 이는 3.2.3의 Step 1에서 보인 규칙에서 max_value와 min_value에 해당하는 값이다.

<표 2> 전체 1순위 신뢰도의 분포

1순위 신뢰도	90이상	80이상	70이상	60이상	50이상	40이상	30이상
총 건수	29	60	181	553	2079	7676	20859

위의 내용에 따라 (그림 15)와 같은 판단 사례를 정한다. 이 때 1순위 항목의 신뢰도가 50%이상이고 1, 2순위격차가 20%이상인 'data_id=634227'의 경우에만 결과를 인정한다. 반면, total score가 현저히 낮은

Rank / Data_id	1	%	2	%	3	%	4	%	5	%	Total Score	Top Rank	Assign	Human Decision
634227	F	97	A2	1.0	B2	1.0	B3	1.0	B1	0.0	726.33	F	F	F
1678389	B3	29	A3	28	A1	17	F	15	B2	10	514.42	B3	I	
27698	B3	28	F	23	B2	17	B1	16	A3	16	287.12	B3	I	F
825942	F	39	A2	20	B3	17	A3	13	B1	10	31.6	F	I	F
											446.07	기존분류	제안방법	

B : 모호
F : 일반
I : 학습

(그림 15) Rank Score와 Total Score를 고려한 판단 예

'data_id=825942' 경우에는 결과를 무시하고 따로 선별하여 고려할 수 있도록 항목을 구분해 놓는다. 본 실험에서는 이러한 문서와 미정항목 B를 갖는 문서에 대해 미정항목 T(training)라고 지정하고 직접 내용을 확인한 후, 목표항목의 학습대상에 추가한다.

■ Step 2 : 항목의 순위차(distance) 고려

이 과정은 위 Step 1에서 미정항목 T로 결정된 문서들에 적용하여 실제에 가까운 결정값을 얻어내기 위한 것이다. (그림 16)과 <표 3>은 3장에서 정의한 계산함수의 실제 적용 예이며 다음과 같이 설명된다. 'data_id=27698'의 경우 미정항목 B3를 기준으로 2순위인 F그룹과 A3그룹에 대해 <표 3>과 같은 계산 예를 보인다. 이 때 결정값은 식 (3)에 의한 *min{D(B3,F),D(B3,A)}* 값을 취해 얻어진 F항목이다. 이 때 피드백 사항은 사용자에게 이 문서를 텍스트 마이닝 분류의 재학습 과정에서 F항목에 추가하여 훈련시키도록 알린다.

<표 3> 항목의 순위차(distance) 계산 예

'data_id'='27698'의 경우 분류를 위한 계산 예:
·D(B3,F) = 1 * 0.15 = 0.15 -> 선택
·D(B3,A3) = 4 * 0.35 = 1.4

이처럼 항목간의 수치와 항목간의 거리차를 함께 고려하면 수치값을 고려했을 때 보다 더 세분화된 기준으로 동작하여 실제에 가까운 결과를 얻어낼 수 있다. 단, 'data_id=26'의 경우와 같이 미정항목 B에 대하여 A와 F 사이의 간격이 같을 경우 거리함수에 의해 A로 결정되나, 이는 오히려 오류로서 동작하게 된다. 따라서, 항목간의 거리가 같을 경우에는 사용자가 직접 확인된 결과값으로 지정해주기로 한다.

■ Step 3 : Step 1과 Step 2의 결과를 이용하여 훈련데이터 생성

위의 각 단계를 진행하는 과정에서 미지정문서에 대한 분류가 점차적으로 목표항목에 가깝게 지정되고 있음을 알 수 있다. 이 과정에서는 데이터마이닝의 교사학습을 위해, 앞서 본 (그림 15, 16)의 수치적 근거사항과 항목간 거리차를 고려한 사례들을 <표 4>와 같이 구축한다. 이 판단사례는 데이터마이닝 분류수행을 위한 학습패턴으로 동작한다.

<표 4> Step 1과 Step2 의 결과를 이용하여 생성한 훈련 데이터

Num	Rank Score / Data_Id	1	%	2	%	3	%	4	%	5	%	Total Score	Assign
1	634227	F	97	A2	1	B2	1	B3	1	B1	0	726.33	F
2	1114467	F	74	B3	11	A3	6	A1	5	B2	3	662.8	F
3	1023579	F	43	B2	35	A1	10	B3	9	A3	2	1717.6	F
4	825942	F	39	A2	20	B3	17	A3	13	B1	10	31.6	T
5	554434	F	28	B3	24	A1	18	B2	17	B1	13	392.33	F
6	27698	B3	28	F	23	B2	17	B1	16	A3	16	267.12	F
7	1678389	B1	29	A3	28	A1	17	F	15	B2	10	514.42	A
8	928687	B2	30	F	25	B3	21	A1	14	B1	12	378.16	F
9	10389	A3	74	F	9	B1	7	A2	6	A1	4	139	A

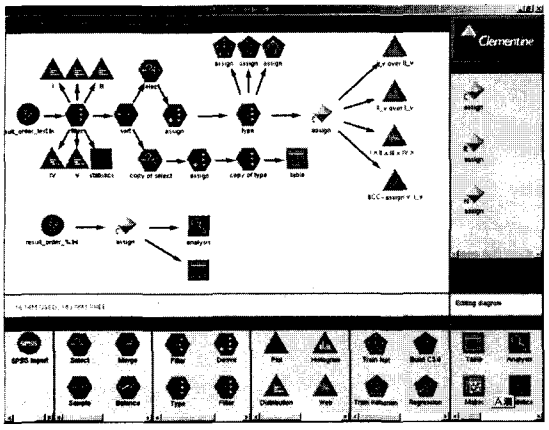
4.4.2 데이터마이닝 분류

앞서 구축한 패턴으로 분류모델을 작성하여 데이터마이닝 분류를 수행한다. (그림 17)은 SPSS사의 Clementine5.2를 이용한 데이터마이닝 분류과정으로써, 분류모델을 생성한 후 전체 데이터에 적용하는 과정을 보이고 있다. 훈련데이터를 결정트리와 신경망 등의 다양한 분류방법을 이용하여 각각의 분류규칙을 생성하고 있으며, 항목별 분포 파악을 위한 통계적기법과 시각적 기법의 적용과정을 보이고 있다.

Data_Id \ Rank	1	2	3	4	5	Assign	Human Decision	Feedback
1678389	B3	A3	A1	F	B2	A	A	T_A
27698	B3	F	B2	B1	A3	F	F	T_F
302586	A1	B1	A2	F	B3	A	A	T_A
26	A3	B2	B3	F	B1	A	F	T_F
Weight	0.02	0.15	0.25	0.31	0.35			

■ 고려사항
- 미정항목 B 그룹과 목표항목간의 거리 계산 : D(B,A),D(B,F)
- D(X,Y) : X와 Y간의 rank 거리차 × weight , weight = log(sqrt(rank+0.1))
- 결정 : min{D(B,A),D(B,F)} 값을 갖는 항목 선택

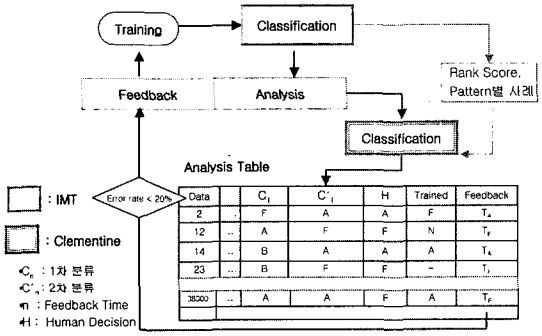
(그림 16) 항목의 순위차(distance)를 고려한 사례



(그림 17) 데이터마이닝을 통한 분류분석(Clementine5.2)

4.4.3 해석 및 피드백

데이터마이닝 분류과정과 이전의 텍스트마이닝 분류 결과로 얻은 사항을 비교하고 해석하여 그 처리사항으로 재학습시키는 과정이다. 이는 TN으로 표기하여 사용자가 해당 문서를 목표항목 N으로 학습시켜야 함을 의미한다. 이 때, 에러율을 줄이기 위해 (그림 18)의 분석 테이블의 변동이 최대한으로 감소할 때까지 재분류를 수행할 수 있다.

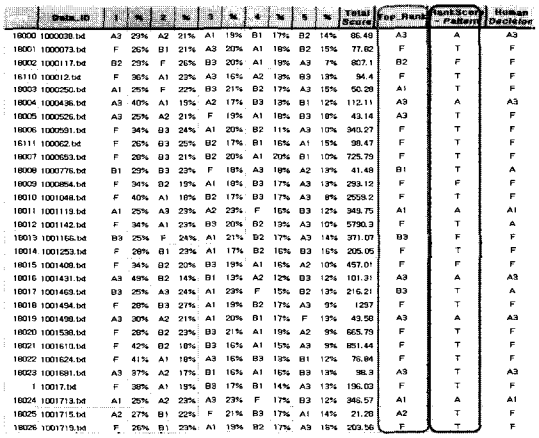


(그림 18) 분류 결과를 비교하여 텍스트마이닝의 학습과정에 피드백

(그림 18)에서 'n'은 피드백 횟수를 의미한다. 즉, C1은 재학습이 1번 이루어진 텍스트마이닝 분류(1차 분류)를 의미하고, C'1은 데이터마이닝 분류(2차 분류)를 의미한다. 'data_id=2'의 경우 1차 분류결과가 F항목이었으나 2차 분류결과는 상반된 A항목이며, 직접 확인된 결과 또한 A항목이다. 이 문서는 이전에 F항목으로 학습된 적이 있었으며 이 때 피드백사항은 T_A로써, 이는 A항목으로 재학습시켜야 함을 나타낸다. 이와 같이, 눈여겨보아야 할 사항은 두 분류 결과의 차이가 심하게 나타날 경우 직접 확인하는 작업이 필요하다는 것이다. 가령, 다음단계에서 목표항목의 학습데이터로 사용되기로 결정된 문서가 이전에 같은 단계에서 이미 학습되었다면, 학습이 불충분함을 의미하므로 학습데이터를 충분히 추가하도록 한다.

4.5 실험 결과

약 38000건의 데이터를 대상으로 제안방법을 실행한 후, 결과가 알려진 400건을 선택하여 검증하여 보았다.(그림 19). 특성벡터의 통계수치에 기반하여 얻은 기존의 텍스트마이닝 분석의 에러율과 본 논문의 제안방법으로 분석(피드백 횟수=1)한 결과의 에러율은 <표 5>와 같다. 목표항목이 2개에 불과하기 때문에 피드백 횟수는 1회로 충분하였다. 이 때, C_0는 피드백 횟수가 0인 텍스트 분류결과를 나타내며 C_1은 피드백 횟수가 1인 텍스트 분류결과를 나타낸다. C'_0 과 C'_1 은 각각 C_0와 C_1의 데이터마이닝 처리 결과를 나타낸다.



	Data_ID	1	%	2	%	3	%	4	%	5	%	Total Score	For_Rank	RankScore_Pattern	Human Decision
18000	1000038.txt	A3	29%	A2	21%	A1	19%	B1	17%	B2	14%	66.49	A3	A	A3
18001	1000073.txt	F	26%	B1	21%	A3	20%	A1	18%	B2	15%	77.82	F	T	F
18002	1000117.txt	B2	29%	F	26%	B3	20%	A1	19%	A3	7%	807.1	B2	F	F
16110	100012.txt	F	36%	A1	23%	A3	16%	A2	13%	B3	13%	94.4	F	T	F
18003	1000250.txt	A1	25%	F	22%	B3	21%	B2	17%	A3	15%	50.28	A1	T	F
18004	1000436.txt	A3	40%	A1	15%	A2	17%	B3	13%	B1	12%	112.11	A3	A	A3
18005	1000526.txt	A3	25%	A2	21%	F	19%	A1	18%	B3	18%	43.14	A3	T	F
18006	1000591.txt	F	34%	B3	24%	A1	20%	B2	11%	A3	10%	340.27	F	T	F
16111	100062.txt	F	26%	B3	25%	B2	17%	B1	16%	A1	15%	98.47	F	T	F
18007	1000653.txt	F	28%	B3	21%	B2	20%	A1	20%	B1	10%	725.79	F	T	F
18008	1000776.txt	B1	29%	B3	23%	F	18%	A3	18%	A2	13%	41.48	B1	T	A
18009	1000854.txt	F	34%	B2	19%	A1	18%	B3	17%	A3	13%	293.12	F	F	F
18010	1001048.txt	F	40%	A1	18%	B2	17%	B3	17%	A3	8%	2559.2	F	T	F
18011	1001119.txt	A1	25%	A3	23%	A2	23%	F	16%	B3	12%	349.75	A1	A	A1
18012	1001142.txt	F	34%	A1	23%	B3	20%	B2	19%	A3	10%	5790.3	F	T	A
18013	1001165.txt	B3	25%	F	24%	A1	21%	B2	17%	A3	14%	371.07	B3	F	F
18014	1001253.txt	F	28%	B1	23%	A1	17%	B2	16%	B3	15%	205.05	F	T	F
18015	1001408.txt	F	34%	B2	20%	B3	19%	A1	16%	A2	10%	457.01	F	F	F
18016	1001431.txt	A3	49%	B2	14%	B1	13%	A2	12%	B3	12%	101.31	A3	A	A3
18017	1001463.txt	B3	25%	A3	24%	A1	23%	F	15%	B2	13%	216.21	B3	T	A
18018	1001494.txt	F	28%	B3	27%	A1	19%	B2	17%	A3	9%	1297	F	T	F
18019	1001498.txt	A3	30%	A2	21%	A1	20%	B1	17%	F	13%	43.58	A3	A	A3
18020	1001538.txt	F	28%	B2	23%	B3	21%	A1	19%	A2	9%	665.79	F	T	F
18021	1001610.txt	F	42%	B2	18%	B3	16%	A1	15%	A3	9%	851.44	F	T	F
18022	1001624.txt	F	41%	A1	18%	A3	16%	B3	13%	B1	12%	76.84	F	T	F
18023	1001681.txt	A3	37%	A2	17%	B1	16%	A1	16%	B3	13%	98.3	A3	T	A3
1	10017.txt	F	38%	A1	19%	B3	17%	B1	14%	A3	13%	196.03	F	T	F
18024	1001713.txt	A1	25%	A2	23%	A3	23%	F	17%	B3	12%	346.57	A1	A	A1
18025	1001715.txt	A2	27%	B1	22%	F	21%	B3	17%	A1	14%	21.28	A2	T	F
18026	1001719.txt	F	26%	B1	23%	A1	19%	B2	17%	A3	15%	209.56	F	T	F

(그림 19) 확인된 결과값에 의한 분류 결과 검증

초기 학습문서는 383건이며 주요내용은 다음과 같다. <표 5>

■ 기존 텍스트마이닝에 의한 분류 결과(C_0) 미분류 문서가 127건으로 오류율은 32%이다. 단, <표 5>에서 F(A)는 실제 A로 지정되어야 할 문서가 F로 지정되어 발생한 오류를 의미하며, B(A|F)는 A 또는 F로 지정되어야 하나 어느 쪽으로도 분류되지 못하여 발생한 오류를 의미한다. 이 때, 30건의 문서가 실제로 A항목이지만

F항목으로 분류되었다.

■ 제안방법의 의한 분류 결과(C_1) 훈련문서는 C'0에 30건과 C_1에 추가된 100건 총 513건이며, 미분류 문서는 총 34건으로 9%의 오류율을 갖는다. 이 때, 31건의 문서가 실제 F항목이지만 A항목으로, 3건의 문서가 여전히 미정항목 B로 지정되었다.

<표 5> 피드백 횟수 = 1 의 실험 결과

●기존 방법과 제안방법의 오류율 비교

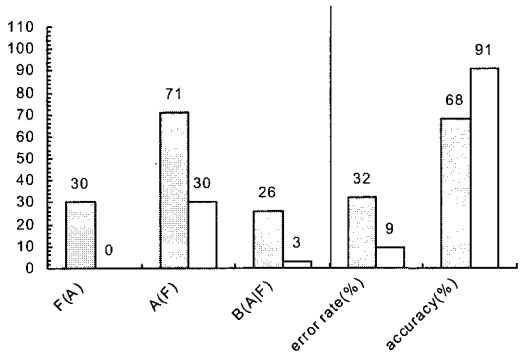
분류방법 \ 오류내용	F(A)	A(F)	B(A|F)	Error rate
기존방법 C_0	30	71	26	0.32 (127/400)
제안방법 C_1	0	31	3	0.09 (34/400)

●기존 방법과 제안방법의 Training 문서의 비교

분류도구 \ feedbank 회수	0			1			총	비고
텍스트마이닝(C)	A	B	F	A	B	F	413	전체 약38000건
	120	119	144	20		10		
데이터마이닝(C')	A	T	F	A	T	F	100	
	21	43	36	-	-	-		
총	483			34			513	

실험결과는 피드백을 통한 적절한 학습데이터의 추가와 처리사항의 반영으로 비교적 적은 학습량에도 불구하고 이전 보다 뚜렷한 향상을 보이고 있다. 기존 텍스트마이닝 분류결과의 정확도가 68%에서 91%로 23%가 상승되어 기존보다 정확도와 품질이 향상되었음을 알 수 있다. 이는 패턴에 의한 세분화된 분류기준이 항목별 근소한 수치를 가진 모호성이 강한 미정문서들의 분류를 가능하게 한다는 것을 의미한다.

(그림 20)은 기존 텍스트마이닝에 의한 방법(C_0)과 제안방법(C_1)에 의한 방법을 적용했을 때의 분류 오류율과 정확도를 비교하고 있으며 음영 부분이 C_0 에 해당한다. C_1 영역은 피드백을 통한 통합 마이닝 시스템을 적용한 후 다시 텍스트마이닝에 의한 재분류를 수행했을 때의 결과를 나타낸다.



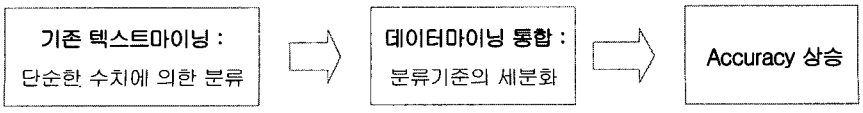
▨ 기존 텍스트마이닝 분류결과에 의한 오류율
□ 제안방법에 의한 텍스트마이닝 분류결과에 의한 오류율(피드백 1회)

(그림 20) 기존방법(C_0)과 제안방법(C_1) 적용시 분류결과의 오류율과 정확도 비교

5. 결론

웹 문서와 같은 비구조적 문서에 대한 분류 작업에 있어서 현재 통용되고 있는 텍스트마이닝 시스템들은 비교적 단순한 방법으로 대규모 문서를 빨리 처리할 수는 있으나 오류율이 비교적 높은 것이 단점으로 지적되고 있다. 반면 자연어처리에 기반한 문서처리의 경우 오류는 줄어들지만 의미파악을 위한 계산이 복잡하고 처리속도가 문제가 되는 경우가 많다. 본 연구에서는 텍스트마이닝의 간편함을 최대한 유지하면서 문서의 내용을 반영할 수 있도록 데이터마이닝의 분류모듈을 이용할 수 있는 방법을 제안하였다. 텍스트마이닝과 데이터마이닝을 결합함으로써 문서분류의 오류를 현저히 줄일 수 있었다. 이러한 방법은 특히 최근 활발히 진행되고 있는 바이오인포매틱스 분야에서 많이 나타나는 대량의 분석 가치가 높은 비구조적인 데이터를 분류하는 작업에 효과적으로 활용될 수 있을 것이다.

그동안 웹 중심의 자연어 텍스트문서를 자동 분석하기 위한 실용적인 시스템을 구축하려는 시도들이 많았으나 그다지 만족스러운 결과를 보여 주지 못하고 있다는 점에서 앞으로도 좀 더 다양하고 새로운 접근이 이



(그림 21) 본 연구의 제안방법을 통한 기대효과

루어져야 할 것이다. 그런 점에서 본 논문에서 제안한 텍스트마이닝과 데이터마이닝의 상호보완적 활용은 처음 시도되는 방법으로서 두 방법이 갖는 장점을 최대한 활용할 수 있는 솔루션이 될 것으로 기대된다.

6. 참고문헌

[1] Kanagasa R. and A-H. Tan, "Topic Detection, Tracking and Trend Analysis Using Self-Organizing Neural Networks". in Proceedings, Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01), Hong Kong, pages 102-107, 2001.

[2] A-H Tan, "Predictive Self-Organizing Networks for Text Categorization". in Proceedings, Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01), Hong Kong, pages 66-77, 2001.

[3] Lakshmi V., A-H Tan, and C-L Tan, Web Structure Analysis for Information Mining. Accepted by ICDAR'01 Workshop on Web Document Analysis, Seattle, September 10-13, 2001.

[4] Mooney J., "Using Information Extraction to Aid the Discovery of Prediction Rules from Text", in Proceedings of the Sixth ACM SIGKDD International Conference on KDD Workshop on Text Mining, pages 51 - 58, Boston, MA, August, 2000

[5] Shankar S., Karypis G., A Feature Weight Adjustment Algorithm for Document Categorization. in Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000.

[6] Mobasher B., Cooley R., Srivastava J., "Automatic Personalization Based on Web Usage Mining", ACM August, 2000.

[7] Baixeries, J., G. Casas, J. L. Balcazar, "Frequent Sets, Sequences, and Taxonomies : New, Efficient Algorithmic Proposals", Technical Report LSI-00-78-R, 2000.

[8] Dorre J., Gerstl P., and Seiffert R., "Text Mining: Finding Nuggets in Mountains of Textual Data", in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999.

[9] Lee Hing Yan, "Text Mining-Knowledge Discovery from Text", Trend in Knowledge Discovery from Databases, 29th June 1999.

[10] Larsen B., Aone C., "Fast and Effective Text Mining Using Linear-Time Document Clustering", in Proceedings of the Fifth ACM SIGKDD International Conference on KDD, 1999.

[11] Cooley R., Mobasher B., and Srivastava, J., "Data Preparation for Mining World Wide Web Browsing Patterns", Journal of Knowledge and Information Systems, 1(1), 1999.

[12] Kevin K., "Mining Online Text", Communications of the ACM 42, Nov 1999.

[13] Turney P., Learning to Extract Key Phrases from Text. Technical Report ERB-1057, National Research Council, Institute for Information Technology, 1999.

[14] Clifton C. and Cooley R., TopCat: Data Mining for Topic Identification in a Text Corpus, in Proceedings of the Third European Conference of Principles and Practice of Knowledge Discovery in Databases, Prague, Czech Republic, 1999.

[15] Yang Y., An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval, 1999.

[16] Platt J., Heckerman, D., and Sahami M., Inductive Learning Algorithms and Representations for Text Categorization, in Proceedings of the Seventh International Conference on Information and Knowledge Management, 1998.

[17] Perkowitz M. and Etzioni O., Adaptive Web Sites: Automatically Synthesizing Web Pages, in Proceedings of Fifteenth National Conference on Artificial Intelligence, Madison, WI, 1998.

[18] E-H Han, Karypis G., Kumar, V. and Mobasher, B., Hypergraph Based Clustering in High-Dimensional Data Sets: a Summary of Results, IEEE Bulletin of the Technical Committee on Data Engineering, (21) 1, March 1998.

[19] Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P., "From Data Mining to Knowledge Discovery", in Advanced in Knowledge Discovery and Data Mining, AAAI Press/MIT Press, pp. 1-34, CA, 1996.

[20] IBM Text Mining, http://www-4.ibm.com/software