퍼지추론과 코호넨 신경망을 사용한 유즈넷 뉴스 필터링

김종완, 조규철, 김병익
대구대학교 정보통신공학부
{jwkim, kccho}@webmail.taegu.ac.kr, smbi@orgio.net

Usenet News Filtering using Fuzzy Inference and Kohonen Network

Jong-Wan Kim, Kyu-Cheol Cho, Byeong-Ik Kim
School of Computer and Information Engineering, Daegu University

요 약

인터넷을 통해 제공되는 많은 양의 뉴스 정보 중에서 찾고자 하는 정확한 정보를 빠른 시간 안에 검색하고, 원하는 정보만 필터링 하는 것이 필요하다. 먼저, 인터넷에 접속된 뉴스서버들의 뉴스 문서를 각 그룹별로 수집한다. 수집된 뉴스 문서를 대상으로 퍼지추론을 통하여 문서를 대표하는 키워드를 추출하여 데이터베이스에 저장한다. 각 뉴스그룹의 문서에서 단어들을 분석하여 입력된 단어들의 개수를 이용하여 정규화시켜서 대표적인 비지도학습 신경망인 코호넨 신경망을 사용하여 학습시킨다. 코호넨 신경망으로 추출된 단어들의 연관성을 활용하여 뉴스그룹을 클러스터링한다. 최종적으로 사용자가 관심있는 키워드를 입력하면, 학습된 신경망이 유사한 뉴스그룹들을 사용자에게 제시해준다.

1. 서 론

1990년대 이후 인터넷이 급속도로 발전하고, 일반 사용자들에게 보급되면서 인터넷을 통해 제공되는 정보의 양도 기하급수적으로 증가하고 있다. 따라서 사용자들은 웹상에서 존재하는 많은 자료들 중에서 찾고자 하는 정확한 정보를 빠른 시간 안에 검색하고, 원하는 정보만 필터링 되어져서 제공받기를 원하고 있다[1]. 하지만 사용자 입장에서 보면 아직 그러한 서비스가 만족스럽게 제공받고 있지 못하다. 특히 인터넷 사용자들이 많이 사용하는 기능 중의 하나인 뉴스 서비스의 경우 매일매일 사용자에게 전달되는 많은 뉴스와 스팸메일을 포함한 광고들 중에서 실제적으로 필요로 하는 뉴스를 검색해 내는 필터링의 기능이 절실히 요구되고 있다[2].

본 논문에서는 수많은 뉴스서버들에서 제공하는 뉴스들 중 사용자가 원하는 정확한 뉴스만을 필터링 해주는 서비스에 대한 사용자 요구를 해결하기 위해 먼저, 인터넷에 접속된 뉴스서버들에 접속해서 뉴스를 모아오도록 한다. 그리고 모아온 뉴스들의 대표 용어들의 추출 방법에서는 우선 뉴스들로부터 후보 용어들을 추출하고 퍼지 추론을 적용하여 대표 용어들을 선택한다. 제안 방법의 성능은 대표 용어들을 선택하는 방법에 의해 영향을 크게 받는다. 따라서 뉴스그룹에서 대표 용어를 추출하는 문제는 불확실성을 내포하고 있으므로 이러한 문제 해결에 효과적인 퍼지 추론을 대표 용어의 선택 방법에 적용하였다. 뉴스그룹을 검색하기 위해 사용자는 미리 관심있는 분야에 대한 키워드를 입력할 수 있고, 시스템이 각 뉴스서버들에서 모아온 뉴스들 중에서 사용자가 입력한 키워드에 맞는 뉴스를 걸러낼 수 있도록 뉴스 필터링 시스템을 구현하였다. 또한 이 시스템에서는 사용자가 입력한 키워드를 통해 사용자의 기호를 학습하여 뉴스를 필터링하기 위해 신경망 기법 가운데 대표적인 비지도 학습 알고리즘인 코호넨 신경망을 이용하였다. 코호넨 신경망은 지속적인 사용자의 피드백을 요구하지 않는 비지도 학습의 한 종류로 사용자가 입력한 키워드만 가지고 뉴스그룹들을 학습시킬 수 있어서, 프로파일을 이용한 뉴스 그룹의 순위를 부여할 수 있다는 장점이 있다. 이에 본 연구에서는 코호넨 신경망을 필터링 알고리즘으로 채택하였다.

2. 시스템 구조와 학습

2.1 시스템의 기본 구조

본 논문에서 구현한 뉴스 필터링 시스템은 자바언어로 구현된 Bigus의 뉴스 필터링 시스템[3]을 참고하여, 사용자 인터페이스를 GUI로 하기 용이한 Swing을 사용하여 자바언어로 구현하였다. 또한 java.net.Socket class를 사용해서 NNTP Server에 접속하였고, NNTP Protocol을 통해서 뉴스그룹을 선택하고, 뉴스문서의 목록 및 내용을 조회할 수 있도록 하였다. 유즈넷 접속과 뉴스그룹, 뉴스문서 조회에 대한 기능을 NewsHost class에 구현하였는데 유즈넷은 news.kornet.net 같은 도메인으로 접속할 수 있는 서버가 있고, 각 서버마다 여러 개의 그룹이 있다. 그러나 존재하지 않는 뉴스그룹이 상당히 많기 때문에 이 프

로그램을 사용하여 뉴스서버에서 각 뉴스그룹에 접속할 경우 그 존재하는 뉴스그룹의 경우에는 서버 응답 메시지의 첫 시작이 "211"로 시작한다. 이것을 이용하여 뉴스그룹의 존재 유무를 판단한다.

뉴스서버에서 뉴스를 읽어올 때 먼저 뉴스의 시작번호와 끝번호를 읽어온 후 그 시작번호부터 끝번호까지 뉴스를 읽어오도록 명령어를 실행한다. 이때 처음에 읽어왔던 시작번호와 끝번호의 정보와는 달리 뉴스가 그만큼 존재하지 않는 경우가 종종 있다. 존재하는 문서일 경우 서버 응답 메시지의 첫 부분이 "223"으로 시작한다. 뉴스그룹의 존재 유무의 판단과 같은 방법으로 문서의 유무도 판단한다.

2.2 대표 용어 선택 방법

뉴스들로부터 사용자의 관심 내용을 가장 잘 대변하는 대표 용어의 선택이 무엇보다 중요하다. 이들 대표 용어들과 각각의 뉴스 내에 존재하는 후보 용어들과의 발생 빈도 유사도 계산이 서로 이루어짐으로 선택 방법과 기준이 성능에 큰 영향을 미친다. 특정 용어의 중요도 계산에 사용되는 입력 정보(예: TF, DF, IDF)들은 정량적으로 정확히 해석될 수 없는 부정확하고 불확실한 특성을 내포하고 있다. 따라서 이러한 불확실성의 문제 해결에 효과적인 퍼지 추론을 적용하여 후보 용어들의 가중치을 계산하고 이 값들에 따라 선택 우선 순위를 부여하였다.

퍼지 추론을 이용한 대표 용어 중요도를 계산하기 위해 뉴스들은 불용어 처리 그리고 Porter stemmer를 사용한 스테밍 과정에 의해 후보 용어들의 집합으로 변형되며, 이 집합으로부터 각각의 용어들의 TF(Term Frequency), DF(Document Frequency), IDF(Inverse Document Frequency) 정보가 구해진다. 이들 정보들이 퍼지 추론을 위한 퍼지 제어의 퍼지 입력으로 이용된다[4]. 퍼지 입력 변수들을 설명하면 아래와 같다.

TF(Term Frequency)

각 용어의 발생 빈도수는 퍼지 계산에 사용되어지기 위해 정규화(NTF) 되어야 하며 아래의 식 (1)을 사용하였다.

$$NTF_i = \frac{\frac{TF_i}{DF_i}}{\underset{j}{Max}\left[\frac{TF_j}{DF_j}\right]} \qquad (1)$$

TF_i : 예제 문서 집합에서 i 번째 단어의 발생 빈도수
DF_i : 예제 문서 집합에서 i 번째 단어를 포함하는 문서의 수

DF(Document Frequency)

각 용어의 예제 문서 집합 내에서의 문서 발생 빈도수를 나타내며 TF와 마찬가지로 아래의 식 (2)을 사용하여 정규화(NDF) 하였다

$$NDF_i = \frac{\frac{DF_i}{TD}}{\underset{j}{Max}\left[\frac{DF_j}{TD}\right]} \qquad (2)$$

TD : 예제 문서의 수
DF_i : 예제 문서 집합에서 i 번째 단어를 포함하는 문서의 수

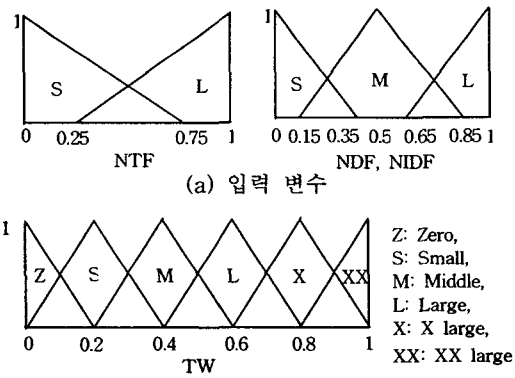
IDF(Inverse Document Frequency)

각 용어의 전체 예제 문서 집합 내에서의 역문헌 빈도수를 나타내며 아래의 식 (3)을 사용하여 정규화(NIDF) 하였다.

$$NIDF_i = \frac{IDF_i}{\underset{j}{Max}\,[IDF_j]} \qquad (3)$$

IDF_i : i 번째 단어의 역문헌 빈도수

그림 1은 퍼지 추론을 위하여 사용된 입출력 변수들의 멤버쉽함수를 나타내고 있다. 용어별로 구해진 NTF, NDF, NIDF 값들을 퍼지 추론에 적합한 형태로 퍼지화 시켜야 한다. 본 논문에서는 그림 1와 같은 삼각형 형태의 퍼지 수를 사용하였다. 그림 1(a)에서 NTF 입력 변수 값은 S(Small)과 L(Large)로 2개의 소속 함수 부분으로 나누었고 NDF 와 NIDF 들은 S(Small), M(Middle), L(Large)로 하였다. 그림 1(b)에서 중요도를 나타내는 퍼지 출력 변수인 TW(Term Weight)는 6개의 소속 함수 부분으로 나누었다.



(a) 입력 변수

(b) 출력 변수

그림 1. 퍼지 입출력 변수

표 1은 NTF 퍼지 입력값의 소속 정도에 따라 두 부분으로 나누어 규칙들을 표현하고 있다. 작성 과정의 예를 살펴보면 NTF 입력값이 S(발생 빈도수가 낮음), NDF는 S(문서 빈도수가 낮음), 그리고 NIDF가 S(역문헌 빈도수가 낮음)일 경우 모든 특성치들이 낮은 값을 가짐으로 중요 용어로서의 관련 정도를 Z(거의 관련 없음)로 두었다. NTF가 S, NDF가 L(문서 빈도수가 높다), 그리고 NIDF가 S 일 경우, 해당 용어가 대부분의 예제 문서들에 등장함으로 인해 관련성을 높게 평가 할 수 있지만 NTF와 NIDF 둘 모두가 낮은 값을 취함으로 관련 정도는 S(낮음)으로 설정하였다. 이와 같은 과정으로 다른 모든 규칙들의 후건부를 설정하였다.

NTF, NDF, NIDF 퍼지 입력값을 위의 결과로 생성된 18개의 추론 규칙별로 이들의 전건부의 소속 함수에 적용시킨다. 각각의 소속 정도가 구해지면 이들 중에서 최소값을 취한다. 그 결과 규칙별로 하나씩의 퍼지 값이 생성되며 이 퍼지 값들을 퍼지 출력 변수 TW에 따라 6개의 그룹으로 분류하고 그룹별로 해당 그룹에 속한 퍼지 값들 중 최대값을 취하여 총 6개의 퍼지 값들을 생성한다. 최종적으로 이들 6개의 퍼지 값들을 무게중심법(center of gravity)으로 비퍼지화(defuzzification)한 값이 해당 용어의 중요도 값으로 결정되어진다.

표 1. 퍼지 추론규칙

NDF \ NIDF	S	M	L
S	Z	S	M
M	S	L	X
L	S	X	XX

NTF = S

NDF \ NIDF	S	M	L
S	Z	Z	S
M	Z	M	L
L	S	L	X

NTF = L

2.3 학습 방법

제시된 뉴스 필터링 시스템은 코호넨 신경망을 이용하여 사용자의 기호를 학습하게 하였다. 먼저, 사용자는 자신이 원하는 뉴스에 포함될 키워드를 입력할 수 있고, 시스템은 각 뉴스문서에 대해서 각 키워드들이 몇 번 나타나는지를 코호넨 신경망에 대한 입력 벡터로 취급해서 학습한다. 코호넨 신경망 학습 알고리즘은 아래와 같이 6단계로 구성된다[4].

[단계 1] 연결강도벡터 W를 초기화한다.

N개의 입력으로부터 M개의 출력 뉴런 사이의 연결강도를 임의로 생성되는 작은 값으로 초기화한다. 이웃반경은 충분히 크게 잡은 후 점차 줄어든다.

[단계 2] 새로운 입력벡터 X를 제시한다.

[단계 3] 입력벡터와 모든 뉴런들 간의 거리를 계산한다. 입력과 출력 뉴런 j 사이의 거리 dj는 다음과 같이 계산한다.

$$dj = \sum_{i=0}^{N-1} [Xi(t) - Wij(t)]^2 \quad (4)$$

[단계 4] 최소거리에 있는 출력 뉴런을 승자 뉴런으로 선택한다. 최소거리 dj 인 출력뉴런 j*를 선택한다.

$$j^* = \min_j dj, \quad j \in \text{출력뉴런} \quad (5)$$

[단계 5] 승자 뉴런 j*와 그 이웃들의 연결강도를 재조정한다. 뉴런 j*와 그 이웃 반경내의 뉴런들의 연결강도를 다음 식에 의해 재조정한다.

$$Wij(t+1) = Wij(t) + \alpha \cdot (Xi(t) - Wij(t)) \quad (6)$$

$$\alpha = \alpha_0 \cdot (1/epoch) \quad (7)$$

여기에서 j는 j*와 이의 이웃반경내의 뉴런이고, i는 0에서 N- 1까지의 정수값이다. α는 0과 1사이의 값을 가지는 이득항(gain term)인데 시간이 경과함에 따라 점차 작아진다. 본 연구에서 α값은 초기값 α_0로 0.9를 사용하였다.

[단계 6] 단계 2로 가서 반복한다.

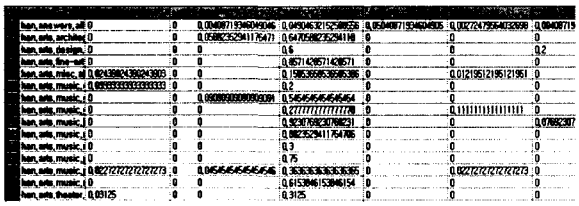
3. 구현 및 실험

먼저, 훈련 데이터(training data)를 모으기 위하여 자바의 Socket Class를 이용하여 NNTP Server (news.kornet.net)에 접속한 후, 각 뉴스그룹에서 뉴스문서를 내려 받았다. 이때 이미 삭제되었거나 옮겨진 뉴스그룹과 10개 이하의 문서를 가지고 있는 뉴스그룹은 제외시켰다.

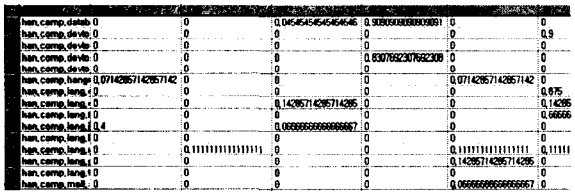
실험 결과 126개의 뉴스그룹을 검색하여 126개 모두를 훈련데이터로 사용하였을 경우와 han.comp로 시작하는 51개의 특정분야의 뉴스그룹을 가지고 실험하였다. 실험을 두가지 방식으로 수행한 이유는 일반적인 뉴스그룹을 대상으로 포괄적인 실험도 수행하고, 제한된 범위의 뉴스그룹을 대상으로 자세한 실험을 수행하여 제안된 방법의 유용성을 보이려고 한다. 출력뉴런의 크기는 126개일 경우에는 5*5이고, 51개의 특정분야의 뉴스그룹에서의 뉴런의 크기는 4*4로 훈련은 각각 1000회 실시하였다. 훈련 데이터는 각 뉴스그룹에서 퍼지 이론을 통해 추출된 단어들을 데이터베이스에 저장해 놓고, 각 뉴스 그룹의 문서에서 단어들을 분석하여 입력된 단어들의 개수를 알아낸다. 본 논문에서 추출된 단어는 126개의 그룹에서는 25개의 단어를 51개의 특정부분의 그룹에서는 17개의 단어를 사용하였다. 각 뉴스그룹의 문서의 수와는 상관없이 단어의 개수만을 파악했을 경우 문서의 수가 많은 뉴스그룹에서

는 대체적으로 단어의 빈도수가 많다. 예를 들어, "han.comp.os.linux.networking" 뉴스그룹의 경우 문서의 수가 1448개인 반면, "han.answers" 뉴스그룹은 24개의 문서만 데이터베이스에 저장되어 있다. 이런 편차를 줄이기 위하여 본 논문에서는 정규화(normalization)를 수행한다[6].

정규화는 (각 단어의 개수)/(뉴스그룹에서 각 단어들이 나타난 총합)으로 계산하여 각 단어들이 뉴스그룹 내에서 나타나는 비율로 한다. 예를 들어, "han.answers"에서 각 단어들이 나타난 총합이 416번이며, "메일"이란 단어는 284번 나타났다. 이 경우에 "han.answers"에서 "메일"이라는 키워드의 비율은 "284/416 = 0.682" 가 된다. 나머지 단어들도 마찬가지 방식으로 계산한 결과가 그림 2, 그림 3과 같다.



[그림 2] 126개의 뉴스그룹의 정규화된 입력벡터



[그림 3] 51개의 뉴스그룹의 정규화된 입력벡터

학습이 끝난 후 각 뉴스그룹의 코호넨 신경망의 출력층 위치와 연결강도 값을 그림 4, 그림 5 같이 데이터베이스에 저장한다. 그림 4는 학습에 사용된 뉴스그룹들이 학습이 완료된 후 2차원 출력층에 배열된 예의 일부를 보여준다. 그림에서 알 수 있듯이, 126개의 뉴스그룹중 코호넨 신경망의 (4,1) 출력 뉴런에 모여 있는 뉴스그룹들은 여러 그룹이 모여 있어 뉴스그룹간의 연관성이 적다. 반면에 51개의 특정부분 뉴스그룹은 코호넨 신경망의(2,2) 출력 뉴런에 모여 있는 뉴스그룹의 연관성이 긴밀함을 알 수 있다.

표 2는 사용자가 입력한 키워드 프로파일을 나타낸 것이다. 사용자가 입력한 키워드를 이용하여 테스트용 입력벡터를 생성한다. 사용자가 입력한 키워드와 미리 입력되어있는 키워드와의 거리를 계산하기 위하여 사용자가 입력하지 않은 키워드의 값을 0으로 하여 입력벡터의 차원을 일치시켰다. 사용자가 입력한 키워드는 각 뉴스 그룹에서 출현한 비율의 평균값을 사용하였다.

han.comp.os.windows.setup.all	4,1
han.comp.security.all	4,1
han.comp.www.misc.all	4,1
han.politics.all	4,1
han.comp.lang.c.all	4,2

[그림 4] 126개의 뉴스 그룹의 일부 학습결과

han.comp.lang.java.all	2,2
han.comp.os.linux.apps.all	2,2
han.comp.os.linux.setup.all	2,2
han.comp.security.all	2,2
han.comp.sys.sgi.all	2,2
han.comp.periphs.input.all	2,3

[그림 5] 51개의 뉴스 그룹의 일부 학습결과

표 2 126개의 뉴스그룹에 사용자 정보

kc	****	조규철	html, http, 서버, 시스템

표 3 51개의 뉴스그룹의 사용자 정보

kc	****	조규철	Linux,OS,SQL,게임

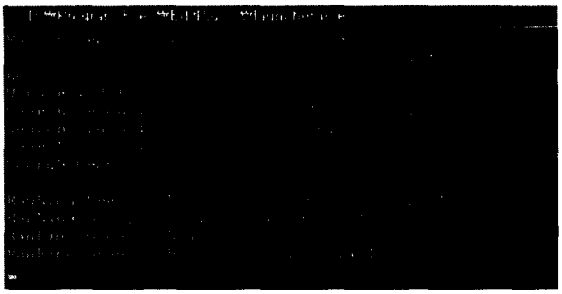
테스트용 입력벡터가 계산되면 코호넨 신경망에 제시하여 가장 가까운 출력뉴런을 선정하고, 이 뉴런에 속하는 뉴스그룹들을 사용자에게 제시한다. 그림 6과 그림 7은 사용자(kc)가 자신의 ID를 입력한 후의 결과 화면으로, 사용자가 입력한 키워드와 미리 학습된 정보를 이용하여 가장 가까운 뉴스그룹을 보여준다. 그림 6에서는 출력뉴런(4,1)이 승자뉴런으로 그림 7에서는 출력뉴런(2,2)이 승자뉴런으로 선정되었다. 본 논문에서는 선정된 승자뉴런과 관련된 뉴스그룹들의 순위(ranking)를 부여하여 사용자에게 순위 순으로 제시한다. 순위는 식 (8)과 같이 (모든 키워드들의 빈도수 비율의 합 / 사용자가 제시한 키워드 수)를 이용하여 계산하였다.

$$ranking(k) = \frac{\sum_{i=1}^{p} f_i(k)}{d} \quad \text{for } all\ k \qquad (8)$$

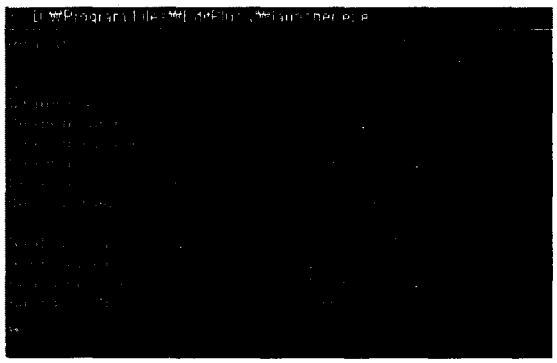
p는 프로파일에 등록된 키워드 수, k는 키워드, d는 사용자가 제시한 키워드의 수를 나타낸다.

예를 들어, 표 5의 han.comp.lang.java.all 뉴스그룹의 경우에 자바 키워드만이 존재하여 ((.4 + 0 + 0 + 0) /4) = 0.1이 된다. 테스트를 위해 사용된 특정 사용자(kc)의

경우에 표 4 및 표 5와 같이 순위가 계산되어, 최종적으로 사용자에게 그림 6 및 그림 7과 같은 결과를 제시한다.



[그림 6] 126개의 뉴스그룹 결과 화면 (사용자 : kc)



[그림 7] 51개의 뉴스그룹의 결과 화면 (사용자 : kc)

표 4 순위 계산 결과

han.comp.os.windows.setup.all	0.1750
han.comp.security.all	0.1625
han.politics.all	0.1176
han.comp.www.misc.all	0.0938

표 5 순위 계산 결과

han.comp.lang.java.all	0.1
han.comp.sys.sgi.all	0.0833
han.comp.os.linux.apps.all	0.0681
han.comp.security.all	0.0625
han.comp.os.linux.setup.all	0.04

표 5에서와 같이 계산이 이루어지면 각각의 그룹의 결과값이 나오는데 'han.comp.os.linux.setup.all'그룹과 같은 경우 "0.05"이하의 작은 값이 계산된다. 본 연구에서는 임계치를 사용하여 임계치 보다 낮은 뉴스그룹은 관련정도가 적다고 판단하여 제외하였다. 현재는 임계치로 0.05을 사용하여, 순위 계산값이 작은 뉴스그룹을 제거시킨후 사용자에게 제시한다.

4. 결론 및 향후 과제

본 연구에서는 사용자가 관심 있는 키워드와 관련 있는 뉴스 그룹을 사용자에게 추천하는 방식으로 유즈넷 뉴스 필터링 시스템을 구현하였다. 뉴스그룹의 문서를 대상으로 퍼지추론을 수행하여 뉴스문서를 대표하는 용어를 추출하였으며, 추출된 단어를 클러스터링하기 적합한 코호넨 신경망으로 학습시켰다.

본 연구의 특징을 다음과 같이 정리할 수 있다. 첫째, 각 뉴스그룹들의 문서의 개수가 서로 달라 비슷한 내용을 지닌 뉴스그룹의 경우라도 문서의 개수가 많은 곳과 적은 곳의 경우 서로간의 단어 빈도수 차이가 많이 나서 거리가 멀어지게되어 비슷한 뉴스그룹으로 분류할 수 없게 된다. 이러한 편차를 줄이기 위하여 정규화를 하였다. 둘째, 테스트시에 입력벡터의 차원을 일치시키기 위하여, 사용자가 입력한 키워드의 경우, 키워드가 나타난 뉴스그룹들의 빈도수의 평균을 구하여 사용하였다. 셋째, 선택된 뉴스그룹을 사용자에게 순위 순으로 제시하여 어떠한 뉴스그룹이 사용자가 입력한 키워드와 가장 유사한 값을 지니고 있는지를 파악할 수 있으며 순위를 부여하지 않고 제시하는 것보다 불필요한 검색을 줄일 수 있다. 넷째, 비슷한 뉴스그룹으로 분류는 되었으나 사용자가 입력한 키워드와 관계가 적은 뉴스그룹들은 사용자에게 제시할 필요가 없으므로 임계치를 사용하여 제거하였다. 다섯째, 퍼지 추론을 통한 뉴스문서로부터 대표용어들을 추출하여 보다 의미있는 필터링 기능을 수행하였다.

향후에는 학습된 뉴스그룹의 문서가 사용자가 원하는 뉴스그룹인지를 피드백받아서 유용성 여부를 판별해야 한다. 또한 각 뉴스그룹의 뉴스 문서를 학습한 후 새롭게 갱신되는 뉴스 문서를 새로운 입력벡터로 사용하여 사용자에게 적당한 문서인지를 파악하여 제공하는 시스템도 추가할 필요가 있다.

참고문헌

[1] 최중민, "인터넷 정보가공을 위한 에이전트 연구동향," 정보처리학회지, 4권 5호, pp 101-109, 1997.

[2] Point CAst Network http://www.pointcast.com/.

[3] Joseph P. Bigus, Jennifer Bigus, Costructing intelligent agents with JAVA, Wiley, 1998.

[4] O, Cordó, F. Herrera, and A. Peregrín, "A Practical Study on the Implementation of Fuzzy Logic Controllers", Intelligent Control, 1998.

[5] 김대수, 신경망 이론과 응용, 하이테크 정보, 1992.

[6] 진승훈, 김종완, 이승아, 김영순, 김병만, "코호넨 신경망을 사용한 유즈넷 뉴스 필터링 에이전트 구현", 산업정보학회논문지, 7권, 5호, 2002