사용자의 지식을 반영한 메일 폴더추천에 관한 방법론

A Knowledge-based Folder Recommendation Procedure for e-mail Classification

류미[a], 박주석[a], 김재경[a]

[a] *School of business Administration, KyungHee University, Seoul, 130-701, Korea*
ubeauty@hanmail.net, jspark@khu.ac.kr, jaek@khu.ac.kr

요약

최근 메일이 커뮤니케이션의 중요한 수단 중 하나로 자리잡고 있으나 과다한 정보 전달 및 원하는 않는 정보의 전달 등으로 인해 사용자가 메일을 확인하고 정리하기 위해 많은 시간과 노력을 투자하고 있다. 본 연구에서는 사용자가 적은 시간과 노력으로 메일을 활용하고, 보다 편리하게 사용할 수 있는 폴더 추천 방법론 개발을 목표로 하고 있다. 이러한 목표를 위해 TF-IDF를 기반으로 하는 다양한 방법론이 개발되고 활용되어 왔으나, 메일이라는 영역의 특성상 단어의 수나 내용에 한계가 있는 경우 안정적인 추천이 이루어지지 못할 수 있었다. 따라서 본 연구에서는 기존의 TF-IDF 방법에 사용자의 지식을 부여한 새로운 방법을 제시함으로써 단어의 수나 내용에 한계가 있는 경우에도 안정적인 추천이 이루어질 수 있도록 하였다. 또한 실제 데이터를 활용하여 기존의 방법과 본 연구에서 제시한 방법론을 비교 실험해 봄으로써, 본 연구에서 제시하고 있는 방법론의 성능을 입증하고자 하였다.

Keywords:

추천시스템, TF-IDF, 사용자 지식(user knowledge), 단어 유사성(keyword affinity)

1. 서론

인터넷의 발달로 인하여 인터넷 사용이 일반화 됨에 따라 메일은 점점 중요한 커뮤니케이션 수단이 되고 있다. 메일 사용자들은 하루에 수십여 통의 메일을 받고 있으며, 메일의 종류는 일반정보, 업무관련, 개인메일에서부터 스팸메일까지 매우 다양하다[2,18]. 그로 인해 수많은 메일 중에서 자신이 필요로 하는 메일을 찾아내기까지 많은 시간과 노력이 필요하게 되었으며, 투자되는 시간과 노력을 줄이기 위해 다양한 연구들이 진행되어 왔다[4,6,7,9,14]. 기존 연구를 살펴보면 스팸메일을 걸러내거나, 사용자가 선호하는 메일을 우선적으로 추천해 주는 등의 연구들이 주를 이룬다 [6,8,18].

메일을 걸러내는 작업 못지 않게 메일을 분류하는 작업에도 많은 시간과 노력이 필요하다. 메일 분류는 각 사용자들의 기준에 따라 메일들을 분류하여 관리하는 것으로써, 기존의 메일관리 시스템들을 보면, 폴더를 생성하여 메일을 조직적으로 관리할 수 있는 기능은 제공하고 있다[15,17]. 그러나 사용자가 직접 폴더를 생성하고 각각의 폴더와 메일을 매칭해야 하므로, 수십여 개의 폴더를 관리하게 될 경우, 신규 메일을 적절한 폴더에 지정해 주는 것에도 많은 노동력과 시간을 필요하다. 이와 같은 이유로 Segal[19,20,21], Kiritchenko[9], Romero[17] 등 여러 연구들을 통해 자동화된 메일 분류 방법에 대한 논의가 이루어져 왔다.

특히 Segal[19,20,21]은 그의 연구를 통해 메일에 포함된 단어를 기반으로 메일을 분류하고 해당 폴더를 추천해주는 SwiftFile을 제시하였다[20]. SwiftFile은 폴더 안에 있는 메일의 단어 빈도를 기반으로 폴더의 프로파일을 생성 관리하여 신규로 도착한 메일이 어디 폴더에 적합한지를 추천해 주지만, 안정적인 추천을 위해서는 100여개 이상의 메일이 있어야 한다[20]. 또한, 사용자의 메일을 관리 패턴이 자주 바뀌는 경우, 신규용어 및 개념이 계속 생성되는 도메인을 포함하고 있는 폴더의 경우 그리고 폴더의 자주 바꾸거나 하는 경우에는 새로운 메일에 대한 폴더추천의 정확성이 떨어질 수 있다. 그러나 이 때, 폴더의 프로파일 생성 및 관리에 있어 사용자 지식을 반영할 수 있다면, 학습을 위해 필요한 메일의 양도 줄일 수 있음으로 인해, 학습에 걸리는 시간을 단축할 수 있을 뿐 아니라 정확도를 높일 수도 있을 것이다.

Lee[11]는 그룹 의사결정 과정에서 사용자의 지식을 이용하여 공통의 의견을 도출하는 방법을 제시하였다. 그들은 유의한 단어들간의 상관관계, 즉 단어의 유사성(Keyword Affinity)을 정의할 때, 사용자의 지식베이스를 이용하여 비슷한 아이디어를 카테고리화 함으로써, 의사결정의 대안을 만들어가는 방법을 제시하였다.

본 연구에서는 Segal[20]이 제시한 SwiftFile을 기반으로, Lee[11]의 단어의 유사성을 이용한 방법을 이용하여 보다 안정적이고 성능이 향상된 메일 추천 알고리즘을 제시하고자 한다. 또한, 기존에 SviftFile을 이용한 방법론과 본 연구를 통해 제시된 방법론을 실제 데이터를 이용하여 비교 실험해 봄으로써, 단어의 유사성이 추천의 성능에 미치는 영향 및 기여도를 측정해 보고자 한다. 본 논문의 구성을 보면, 먼저 2장에서 관련연구와 이론들에 대해 간단히 살펴보고자 한다. 다음으로, 3장에서는 세부적인 알고리즘에 대해 살펴볼 것이며, 4장에서는 실제 이메일 데이터를 이용하여 실험하고자 한다. 마지막으로 5장에서는 본 연구의 결론과 향후 연구 방향에 대하여 논하고자 한다.

2. 문헌연구

2.1 추천시스템

추천시스템은 일반 상품 관련 추천에서부터 영화, 음악, 뉴스, 웹페이지 관련 추천까지 여러 분야와 관련해서 다양한 연구가 진행되어 왔다[8,16]. 활용되고 있는 기법은 크게 내용기반 필터링과 협업필터링으로 나누어 볼 수 있으며, 최근 두 기법을 모두 활용한 혼합형 필터링 기법들에 대한 연구도 다양하게 이루어지고 있다. 본 연구에서 활용하고자 하는 기법은 내용기반 필터링[3]에 하나로써, 필터링 과정에 대해 살펴보면 크게 3단계, 데이터 표현과 프로파일(Data Representation & Profile), 유사성(Similarity), 적합성 피드백(Reference feedback)으로 나누어 볼 수 있다.

- **데이터 표현과 프로파일**: 이메일은 짧은 문장으로 구성된 문서로서 일반문서와 비슷한 성격을 가지므로, 기존의 문서 추천시스템을 기반으로 한다[5,12,13]. 문서 추천에 있어서 문서를 표현하는데 가장 많이 쓰이는 기법은 TF-IDF이다. TF-IDF는 가장 보편화된 방법으로써 문서에 포함된 각 단어를 단어의 빈도와 역문헌 빈도의 곱을 이용한 가중치로써 나타낸다[1]. SwiftFile 또한, 단어의 가중치를 산정할 때 TF-IDF 기법을 사용하였는데, SwiftFile에서는 TF-IDF를 이용하여 사용자의 프로파일을 생성하고, 이 프로파일을 기반으로 새로운 문서가 속할 폴더를 추천하였다. 앞서 말한 바와 같이 새로운 문서가 속할 폴더를 추천하는 것은 프로파일을 기반으로 이루어지므로 프로파일 생성은 추천에 있어 핵심적인 과정이다. 따라서 프로파일 생성에 대해 다양한 연구[10,14]가 이루어져 왔는데, Tsvi Kuflick and Peretz Shoval[10]의 연구에서는 프로파일 생성 방법에 대해 사용자가 직접 생성하는 것(User-Created Profile), 시스템을 활용하여 자동적으로 생성하는 것(System-Created Profile by Automatic Indexing), 사용자와 시스템을 병행하여 활용하는 것(System-plus User-Created Profile), 인공신경망의 학습을 기반으로 생성하는 것(System-Created profile based learning by artificial Neural-Network), 사용자 집단의 특성을 반영하여 생성하는 것(User profile inherited from a User-streotype), 규칙을 기반으로 한 필터링(Rule-based Filtering)으로 나누어 보기도 했다. 또한, Michael J. Pazzani[14]는 메일을 필터링 할 때 어떤 프로파일이 더 효과적인가를 연구하기도 했다.

- **유사성**: 문서의 속성을 잘 표현해 주는 것 중 하나가 벡터공간모델(Vector Space Model)이다. 이 모델에서 사용자의 프로파일은 문서에 대한 사용자의 선호도를 통해 나타나는데, 새로운 문서가 들어오게 되면 기존 사용자 프로파일, 즉 사용자 선호도와 새로운 문서와의 유사성을 분석하여 유사한 것은 추천하고, 유사성이 떨어지는 것은 필터링 한다. 사용자의 프로파일과 신규 문서와의 유사성을 측정하는 기법으로는 피어슨 상관계수, 코사인 등이 있다. SwiftFile에서는 코사인 유사계수인 SIM4를 이용하여 문서 간의 유사성을 구하였으며, 본 연구에서도 코사인 유사계수를 이용하여 문서 간의 유사성을 측정하고자 한다. SIM4를 사용할 경우, $F(M,w)$ 는 신규 메일 M에 속해있는 단어 w의 빈도를 의미하며, $W(f,w)$ 는 폴더 f에 속한 단어 w의 가중치를 의미하는 것으로써, 신규 메일과 기존 폴더의 유사성을 구하는 공식은 (1)과 같다.

$$SIM_4 = \frac{\sum_{w \in M} F(M,w) W(f,w)}{\min(\sum_{w \in M} F(M,w), \sum_{w \in M} W(f,w))} \quad (1)$$

- **적합성 피드백**: 사용자의 프로파일을 생성하는 방법과 사용자의 프로파일을 업데이트하는 방법에 따라서도 추천의 성능은 달라질 수 있다. 적합성 피드백은 최적의

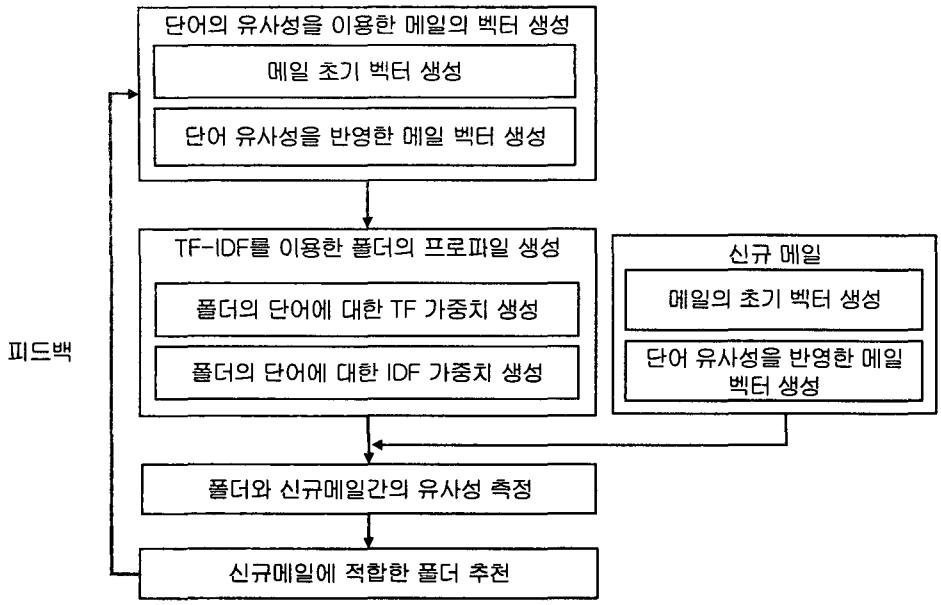

그림 1 - 폴더 추천에 관한 절차

추천분야에서의 적합성 피드백은 추천된 아이템에 대해 사용자가 선호 정도를 반영해주는 것으로, SwiftFile의 경우에서도 폴더에 새로운 메시지가 추가되었을 때 그 폴더에 속한 단어의 가중치를 변경해 준다. 예를 들어 만약, 폴더에서 문서가 삭제되면 기존의 문서가 가지고 있던 단어의 가중치를 빼주게 되는 것이다(2).

$$F(F,w) \leftarrow F(F,w) + F(M,w) \qquad (2)$$

2.2 단어의 유사성을 이용한 사용자 지식 반영

Lee[11]의 연구에서는 아이디어 정리(Idea Categorization)를 위해 사용자의 지식 반영한 방법론을 제시하였다. 회의에서 구성원이 내놓은 각각의 아이디어를 비슷한 것끼리 묶어주기 위하여 키워드에 대한 사용자의 지식을 반영하는 방법을 사용하였다. 아이디어는 단어벡터(word vector)로 (3)과 같이 표현될 수 있다.

$$I_k = (a_{k1}, a_{k2}, ..., a_{kn}) \qquad (3)$$

a_{ki} 는 아이디어 K에 있어서 단어 i의 가중치 값으로, 단어 i가 존재하면 1이고 아니면 0이라고 한다. 아래의 지식베이스(knowledge base)를 보면 단어 T_1이 있을 때, T_1 과 T_2 그리고 T_3 사이에는 각각 0.8과 0.6의 관계가 존재하는 것으로 나타난다.

예로, Idea의 초기 프로파일이 $(T_1, T_2, T_3, T_4, T_5....T_n)$ = (1, 0, 1, 0, 1,...1) 이면, Keyword T2의 경우 T_1과 T_5는 0.8과 0.4의 관계가 있으며, 이때 가장 높은 값을 단어의 유사성을 반영하면 프로파일은 (T_1, T_2, T_3, T_4, T_5....T_n) = (1, 0.8, 1, 0, 1,...1)으로 변경된다.

본 연구에서는 폴더의 프로파일 생성 시 사전에 정의된 사용자 지식을 반영한다. 학습량이 많아지고, 학습기간이 길어지면 모든 추천의 정확성은 안정적일 수 있으나. 초기 학습기간에는 사용자 지식을 이용함으로써 추천의 정확성을 높일 수 있을 것이라 판단된다.

3. 연구 방법론

3.1 개요

본 연구는 신규메일이 도착 했을 때, 그 메일이 어떤 폴더에 적합한지를 분석하고 추천하기 위한 방법론을 제시하고자 한다. 폴더 추천은 일반 추천과 유사한 프로세스로 진행되며, 폴더 추천의 프로세스는 그림 1과 같다. 새롭게 받은 메일이 속할 폴더를 추천하기 위해서는 먼저 각 폴더의 프로파일을 생성해야 한다. 각 폴더의 프로파일은 폴더에 속한 메일들의 단어 벡터를 이용한 TF-IDF 값으로 이루어지게 된다. 새로운 메일이 도착하면, 사용자의 지식을 이용하여 단어 간의 유사성을 정의하고, 유사성이 반영된 메일 벡터를 생성하여, 기존의 각 폴더들의 프로파일과 유사성을 비교해 봄으로써 추천할 폴더를 정의한다. 추천할 때는 새로운 메일과 각각의 폴더간의 유사성을 토대로, 유사성이 가장 높은 폴더의 순으로 상위 3개를 추천한다[20]. 추천한 폴더가 맞으면 사용자가 그대로 추천한 폴더를 선택해서 메일을 정리하고, 틀리면

메일에 적합한 폴더를 찾는다.

메일 필터링이 끝나고 나면 기존 폴더의 프로파일을 업데이트 해준다. 만약에 폴더에 메일이 추가되면 메일에 속한 단어에 대한 가중치를 더해주고, 폴더에서 메일이 제거되면 단어에 가중치를 빼준다. 폴더를 새로 생성하는 경우에는 사용자는 폴더에 속한 단어 관련 지식을 추가할 수 있고, 생성된 이후에는 폴더 프로파일 계산시 사용자 지식이 반영된다. 기존의 폴더에서 사용자의 지식이 추가되거나, 삭제된 경우에는 폴더의 프로파일은 변경사항을 반영하여 새로운 프로파일을 생성한다.

3.2 사용자의 지식을 이용한 메일 표현

폴더의 프로파일을 형성하기에 앞서 먼저 각 메일에 속해있는 단어를 이용하여 각 메일을 벡터의 형식으로 표현해야 하는데, 본 논문에서는 특히 단어의 유사성을 반영하여 벡터를 구하게 된다. 단어의 유사성을 반영하게 되면, 추천을 위한 모델의 학습 효과가 높아지고, 안정적인 추천이 이루어지게 된다. 새로운 메일이 속할 폴더를 추천하고자 할 때, 초기 프로파일을 기반으로 추천이 이루어지는데, 단어의 유사성을 반영할 경우 학습에 사용될 단어의 수가 증가됨으로써 안정적으로 추천할 수 있게 된다. 또한 새로운 단어가 생성될 경우 기존의 방법에 의하면, 어느 폴더에 속할 것인지 정의하기 어려우나, 단어의 유사성을 반영하게 되면, 그 단어가 속할 도메인을 정의해 줄 수 있다.

사용자가 받은 메일 M에 속한 단어 w의 각 가중치는 $M_w = (W_1, W_2, ..., W_n)$ 과 같이 벡터의 형식으로 표현되어지며, 사용자는 자신의 지식을 이용하여 각 단어에 대한 단어의 유사성을 0~1사이의 값으로 지정한다.

메일 M은 n 개의 단어로 구성되어 있으며, W_n 은 메일 M의 n번째 단어에 대한 빈도수, 즉 가중치를 의미한다. 메일 M의 내용 중에서 불용어를 제외한 모든 단어는 벡터(vector)로 표현된다. 폴더는 많은 메일을 주제별 또는 포함된 의미별로 정리하여 향후 필요한 메일을 찾을 때의 신속성을 높이는데 활용된다. 그러나 폴더에 여러 주제가 포함되어 있거나 특색이 없이 일반적으로 사용되는 폴더인 경우, 또는 특정주제를 담는 폴더라도 새로운 개념이 꾸준히 나온다고 할 경우 기존의 메일에 표현되는 단어만 가지고 폴더를 추천하기에는 정확성에 문제가 있을 수 있다.

본 연구에서는 폴더가 가지는 주제나 의미에 대해 사용자가 직접 지식을 부여하고자 한다. 지식베이스는 사용자가 정의한 단어의 유사성이 값으로 표현되어 있다. 해당 폴더의 메일을 표현할 때는 그 폴더에 속해있는 단어들 간의 유사성을 고려하여 표현한다. 예를 들어, 표 1과 같이 CRM폴더에 속한 메일들의 word의 빈도가 나타난다고 가정하자. 이 폴더에 속한 articles과 폴더에는 속해있지 않지만 유사한 단어라고 판단되는 Paper에 대해 사용자가 0.9의 유사성이 있다고 정의하였다면, CRM 폴더의 초기값은 표 2와 같이 수정된다. 즉 1번 메일에서 Paper는 나오지 않았지만 Articles이 3번 나왔으므로 Paper의 빈도수는 3*0.9로 2.7이 되는 것이다.

표 1 - 초기 메일에 속한 단어의 빈도

폴더	메일	CRM	Industry	Articles	Organi-zations	Finance	Manu-facturing
CRM	1	2	1	3	1	2	0
	2	1	0	3	0	0	2
	3	2	1	2	1	0	2

표 2 - 사용자 지식을 이용한 CRM폴더의 단어 빈도

폴더	메일	CRM	Industry	Articles	Organi-zations	Finance	Manu-facturing	Paper
CRM	1	2	1	3	1	2	0	2.7
	2	1	0	3	0	0	2	2.7
	3	2	1	2	1	0	2	1.8

3.3 폴더의 프로파일 생성

폴더의 프로파일은 내용기반추천(Contents-based recommendation)에서의 사용자 프로파일과 유사한 개념으로 볼 수 있다. 먼저 사용자 프로파일에 대해서 살펴보면, 문서 기반의 내용을 추천할 경우 사용자는 추천된 문서나 아이템에 대한 관심의 정도를 암묵적으로나 명시적으로 표현하게 된다. 프로파일을 관리하는 시스템에서는 이러한 사용자의 암묵적, 명시적 평가를 계속 누적하여 사용자의 특성이나 선호도를 형성하게 되며, 이를 사용자 프로파일이라 부른다. 이렇게 형성된 사용자 프로파일은 추천 결과에 많은 영향을 미치게 되므로, 프로파일 생성에 관해 다양하고 지속적인 연구가 이루어지고 있는 것이다.

폴더의 프로파일도 앞서 설명한 사용자 프로파일과 유사한 개념으로, 폴더 마다 각 폴더의 주제에 적합한 메일들이 꾸준히 누적된다. 폴더 프로파일은 폴더에 속한 메일들에 대한 각 요소들의 합이라고 할 수 있으며, 폴더 프로파일과 같은 문서 기반의 프로파일을 생성하기 위해서는 TF-IDF 방법이 주로 사용된다. TF-IDF는 특정 단어의 빈도수와 특정 단어가 나타나는 문서의 빈도수를 이용하여 각 단어에 대한 가중치를 구함으로써, 그 단어가 해당 도메인의 특징이나 성격을 얼마나 잘 반영하였는가를 나타내는 방법이다. TF는 특정

폴더에 속해있는 단어의 가중치를 의미하고, IDF는 그 단어가 다른 폴더에도 포함이 되어 있는지에 대한 가중치를 의미한다.

TF-IDF 값을 구하기 위해서는 먼저 폴더의 단어의 빈도로 구성된 벡터 값을 구해야 하는데, 폴더의 단어 벡터는 다시 폴더에 속해있는 각각의 메일에 대한 단어 벡터의 합에 의해 구해진다. 본 논문에서는 특히 단어 벡터를 구할 때, 단어의 유사성을 반영하는데, 이는 기존 연구를 통해 보여지는 salton and McGill(1993)의 방법이나 Richard and Jeffrey의 방법과는 다른 차이점이다. 메일을 벡터로 표현하는 과정에서 단어의 유사성을 반영한 후, TF-IDF 값을 계산하여 폴더의 프로파일을 형성하게 된다.

폴더를 f, 메일을 M, 메일에 속한 단어를 w라고 할 때, 단어 w의 TF 값을 구하기 위해서는 먼저 폴더 f 에 속한 메일들로부터 단어 w의 빈도수를 구해야 하는데, 그 식은 (4)와 같다.

$$F(f,w) = \sum_{M \in f} F(M,w) \tag{4}$$

폴더 f 에 속해 있는 단어 w의 빈도수를 이용하여, 단어 w가 폴더 f 내에서 차지하고 있는 비중, 즉 가중치 $FF(f,w)$ 를 구할 수 있는데, 그 식은 (5)와 같다.

$$FF(f,w) = \frac{F(f,w)}{\sum_{w' \in F} F(f,w')} \tag{5}$$

마지막으로 $FF(f,w)$ 를 이용하면 특정 사용자의 전체 메일 A 에서의 단어 w의 가중치, 즉 TF 값을 구할 수 있으며, 그 식은 (6)과 같다.

$$TF(F,w) = FF(F,w) / FF(A,w) \tag{6}$$

단어 w에 대한 TF를 구한 다음, w의 IDF를 구해야 하는데, IDF를 구하기에 앞서 단어 w의 $DF(w)$ 를 먼저 구해야 한다. $DF(w)$ 는 전체 폴더 중에 특정 단어가 한번이라도 출현한 폴더 개수를 나눈 값이며, 이 값을 기반으로 단어 w의 중요도, 즉 $IDF(w)$ 가 구해진다. $IDF(w)$ 를 구하는 방법을 식으로 나타내면 (7)과 같다.

$$IDF(w) = \frac{1}{DF(w)^2} \tag{7}$$

위의 내용을 예로써 설명하면 다음과 같다. 먼저 가상으로 CRM 폴더와 Spam 폴더 두 개를 생성한다. CRM 폴더에는 (Table 3-2)와 같이 3개의 메일이 있고, CRM, Industry, Articles, Organization, Finance, Manufacturing, Paper라는 단어가 있으며, 각 단어의 빈도는 5,2,8,2,2,4이라 가정한다면, 폴더 내의 전체 단어의 빈도수의 합은 23이 된다.

다음으로, Table 3-3과 같이 스팸 폴더에 속한 2개의 메일에는 Finance, AD, Shopping, Present, love라는 단어가 있고, 각 단어의 빈도는 4,5,3,3,3 이라고 가정한다. 그러면, 스팸 폴더내의 전체 단어의 빈도수의 합은 18이 된다.

이 때, CRM 폴더내의 Industry라는 단어에 대한 TF는 CRM 폴더 내에서의 Industry 라는 단어의 빈도 2를 CRM 폴더내의 전체 단어의 빈도수의 합 41(23+18)으로 나누고, 그 값을 전체 폴더에 포함되어 있는 industry 단어의 빈도수를 합한 값, 즉 2를 전체 폴더의 단어의 빈도수를 합한 값으로 나누면 된다. 세부적인 계산 과정에 대한 예와 결과 값을 살펴보면 표 3 및 표 4와 같다.

표 3 - 스팸폴더

메일	Finance	AD	Shopping	Present	Love
4	2	2	2	1	1
5	2	3	1	2	2
$F(f,w)$	4	5	3	3	3

표 4 - CRM 폴더의 FF(F,w), FF(A,w) 및 TF 가중치

Weight	CRM	Industry	Articles	Organizations	Finance	Manufacturing	Paper
$FF(f,w)$	0.17	0.07	0.26	0.07	0.07	0.13	0.24
$FF(A,w)$	0.10	0.04	0.17	0.04	0.12	0.08	0.15
TF	1.60	1.60	1.60	1.60	0.53	1.60	1.60

다음으로 각 단어들에 대한 IDF를 구한 후 TF를 고려하여 최종적인 결과 값, 즉 TF-IDF를 구하고자 한다. 각 단어들에 대한 IDF 값을 구하는 것에 대한 예를 보면 다음과 같다. 먼저 표 2를 살펴보면, Finance를 제외한 다른 단어들은 CRM 폴더에만 존재함으로 DF가 1/4 인데 반해, Finance라는 단어는 두 개의 폴더에 다 존재한다. 따라서 DF가 1이 되고, 표 5를 보면 알 수 있듯이, CRM 폴더 내의 다른 단어들은 IDF 4가 된다. 마지막으로 각 단어들의 TF와 IDF의 곱을 이용하여 단어들의 가중치를 구하게 되는데, 그 결과 값은 표 6과 같다.

표 5 - CRM 폴더의 IDF 가중치

Weight	CRM	Industry	Articles	Organizations	Finance	Manufacturing	Paper
IDF	4	4	4	4	1	4	4

표 6 - CRM 폴더의 TF - IDF 가중치

Weight	CRM	Industry	Articles	Organizations	Finance	Manufacturing	Paper
TF	1.60	1.60	1.60	1.60	0.53	1.60	1.60
IDF	4	4	4	4	1	4	4
TF - IDF	6.38	6.38	6.38	6.38	0.53	6.38	6.38

3.4 신규 메일과 폴더 간의 유사성 측정

새로운 메일이 도착했을 때, 어떤 폴더에 속하는 것이 적당할 것인가를 추천하기 위해서는 신규 메일과 각 폴더와의 유사성을 계산해 보면 알 수 있다. 폴더의 프로파일과 신규 메일의 유사성을 계산하기 위해서는 TF-IDF 방법을 통해 신규메일에 속한 각 단어들의 가중치를 구한 후, 폴더에 속한 메일의 단어들과의 유사성을 구해야 한다. 본 연구에서는 SIM4(4) 라는 코사인 거리를 통해 단어 벡터간의 유사성을 구하고자 한다[1].

$$SIM_4(M,f) = \frac{\sum_{w \in M} F(M,w)W(f,w)}{\min(\sum_{w \in M} F(M,w), \sum_{w \in M} W(f,w)} \quad (8)$$

예를 들어, 표 7과 같은 단어 벡터를 가지고 있는 신규메일이 도착했을 때, 먼저 표 8과 같이 신규메일에 속한 단어들에 단어의 유사성을 반영한다. CRM 폴더에 대한 사용자의 지식을 반영한 Paper와 Articles간에 단어의 유사성 값 0.9를 반영하면 단어의 벡터는 표 8과 같은 값이 나타난다.

표 7 - 신규 메일에 대한 단어의 빈도

가중치	Articles	Finance	Vendor	Paper	White	Case	Studies
F(신규 메일)	0	1	1	2	2	1	1

표 8 - 사용자 지식을 반영한 신규 메일에 대한 단어의 빈도

가중치	Articles	Finance	Vendor	Paper	White	Case	Studies
F(신규 메일)	1.8	1	1	2	2	1	1

단어의 유사성을 반영하는 이유에 대해서는 앞서 설명한 바 있다. 그 예로써, 신규로 도착한 메일과 CRM폴더, 그리고 스팸폴더에 대하여, 사용자 지식을 반영한 후 프로파일을 생성하고 유사성을 구한 것과 그렇지 않은 경우의 유사성을 구한 것에 대한 차이점을 비교해 보면 표 9와 같다. 사용자의 지식을 반영하지 않은 경우에는 신규메일은 스팸폴더와 더 가까운 것으로 나타났지만, 반영한 경우에는 CRM 폴더가 유사성이 훨씬 높게 나오는 것을 볼 수 있다. 즉, 단어의 유사성을 반영해야 보다 정확한 추천이 이루어질 수 있음을 알 수 있다. 단어의 유사성은 새로운 메일이 지속적으로 유입됨에 따라, 계속 업데이트 된다(2)

표 9 - CRM 폴더와 신규 메일 간의 유사성

	SwiftFile	SwiftFile + 단어의 유사성 적용
CRM 폴더	0.22	2.53
Spam 폴더	0.28	0.18

4. 실험

실험은 2명의 사용자를 대상으로 하여, 사용자의 메일계정에 관리되는 메일을 사용하였으며, 메일의 개수는 총 586개가 사용되었다. 또한 각 사용자별로 메일을 분류하고, 분류된 기준에 따라 메일을 추천하기 위해, 메일 종류 별로 16개의 폴더를 만들어 정리하였다. 표 10을 살펴보면 실험에 사용된 사용자 별 메일의 개수와 사용자에 의해 정의된 폴더에 대해 정의되어 있으며, 사용자 별 폴더의 특성에 대해서도 설명되어 있다. 먼저, 사용자 1의 경우 총 메일의 개수는 268개이며, 사용자가 정의한 폴더의 수는 16개, 각 폴더에 포함된 메일은 평균 약 17개 정도 되고, 단어의 유사성의 개수는 약 6개 정도 된다. 폴더에 대한 사용자의 지식은 유사 단어의 개수에 의해 표현되는데, 사용자는 자신의 지식을 이용하여 0~1까지 다양한 비율로 각 단어에 대한 유사성을 정의한다.

표 10 - 실험 대상 자료 및 폴더관리 특성

사용자	메일 수	폴더 수	폴더 별 평균 메일 수	폴더별 평균 유사 단어 수
사용자 1	268	16	17	6
사용자 2	318	16	20	9

또한 사용자 2의 경우 총 메일의 개수는 318개이며, 사용자가 정의한 폴더의 수는 16개, 각 폴더에 포함된 메일은 평균 20개 정도 되고, 유사 단어 개수는 약 9개 정도 된다. 본 연구를 통해, 초기 메일의 양이 적을 경우 사용자 지식이 추천의 정확성에 미치는 영향에 대해 살펴보고, 0~100%까지

다양한 비율로 단어의 유사성을 적용해 봄으로써, 사용자 지식을 부여한 정도가 추천성능에 미치는 영향에 대해 살펴보고자 한다. 실험을 위한 모형 구축을 위해 데이터의 70%는 실험용으로 사용하고, 30%는 테스트용으로 사용하였다.

다음은 사용자 지식이 추천 알고리즘의 초기 학습에 얼마나 영향을 미치는지에 대한 실험 결과이다. 실험 결과는 추천 개수에 따라 차이가 있는지 살펴보기 위해 1개를 추천했을 경우와 3개를

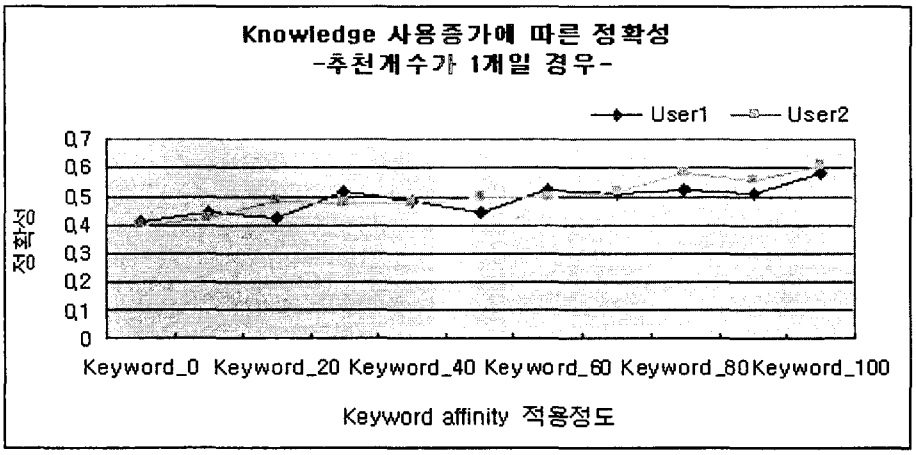

그림 2 - 추천메일의 개수가 1개일 경우 추천성능

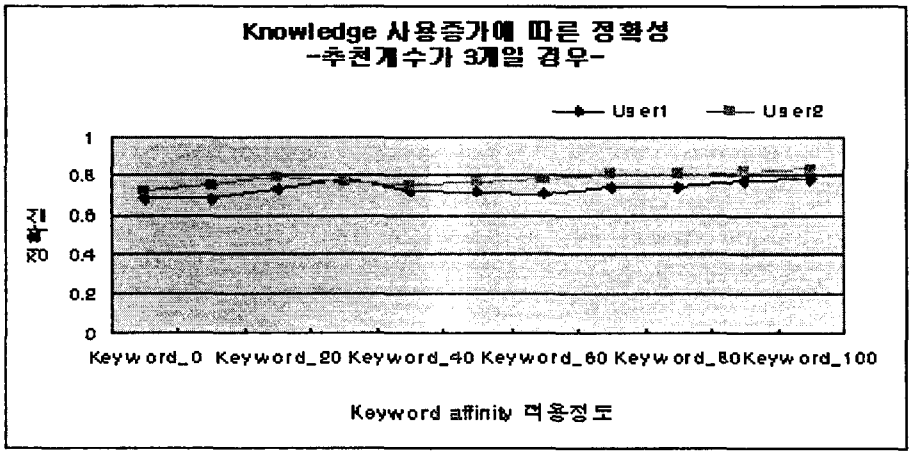

그림 3 - 추천메일의 개수가 3개일 경우 추천성능

추천했을 경우로 나누어 보았다. 먼저 그림 2를 보면 단어 유사성 증가 정도, 즉 사용자 지식의 사용 정도에 따라 한가지 폴더만을 추천했을 경우, 추천 성능을 확인해 볼 수 있다. 그림 2의 결과를 살펴보면 단어의 유사성을 사용하지 않을 경우 추천의 정확성은 0.40 정도이나, 단어의 유사성을 100% 반영했을 경우 정확성은 0.60 정도로 향상된다는 것을 알 수 있다. 또한 전반적으로 볼 때, 단어의 유사성 반영 비율이 증가할수록 정확도가 증가하는 것으로 나타나고 있다.

다음으로 메일이 속하게 될 폴더를 3개 추천했을 경우, 단어의 유사성 적용에 따른 추천 성능의 차이에 대해 살펴보았으며, 그 결과는 그림 3과 같다. 전체적으로 보았을 때, 단어의 유사성 적용 비율이 높아짐에 따라 추천의 성능이 좋아지는 것을 확인해 볼 수 있었으며, 사용자의 지식을 사용하지 않을 경우 추천의 정확성은 0.70인 것으로 나타나고 있으나, 100% 반영한 경우 정확성은 0.81로써 0.1 정도 성능이 향상되었다.

앞서 한 개의 폴더 만을 추천했을 경우와 세 개의 폴더를 추천했을 경우를 비교해 보았을 때, 단어의 유사성 적용 여부에 따른 정확도의 차이는 한 개의 폴더 만을 추천했을 경우 더 큰 것으로 나타나고 있다. 즉, 추천할 폴더가 제한적일 수록 기존에 사용되었던 SwiftFile의 방법보다 단어의 유사성을

적용했을 경우 더 정확한 추천을 할 수 있음을 알 수 있다.

5. 결론

과거의 전통적인 통신수단인 우편, 전화 등과 함께 최근에는 메일이 커뮤니케이션의 중요한 수단 중 하나로 자리잡고 있다. 그러나 과다한 정보 전달 및 원하는 않는 정보의 전달 등으로 인해 사용자가 메일을 확인하고 정리하기 위해 많은 시간과 노력을 투자하고 있어, 그에 따른 사용자의 불편함과 문제점이 심각해지고 있다. 따라서 본 연구에서는 사용자가 적은 시간과 노력으로 메일을 활용하고, 보다 편리하게 사용할 수 있게 하기 위해, 자동으로 메일이 속할 폴더를 추천해 주는 방법론 개발을 목표로 하고 있다. 기존에도 이러한 목표를 위해 TF-IDF를 기반으로 하는 다양한 방법론이 개발되고 활용되어 왔으나, 메일이라는 영역의 특성상 단어의 수나 내용에 한계가 있는 경우 안정적인 추천이 이루어지지 못할 수 있었다. 따라서 본 연구에서는 기존의 TF-IDF 방법에 사용자의 지식을 부여한 새로운 방법을 제시함으로써 단어의 수나 내용에 한계가 있는 경우에도 안정적인 추천이 이루어질 수 있도록 하였다. 다시 말해, 문서 기반의 TF-IDF 방식은 좋은 추천 성능을 보이지만, 추천 성능이 안정화되기까지는 충분한 단어의 수와 내용이 축적되기까지 긴 학습기간이 필요하다. 따라서 이러한 단점을 극복하여 보유한 메일의 양이 작은 경우나, 새로운 용어가 자주 생성되는 도메인의 경우에도 안정적이고 정확한 추천이 이루어질 수 있도록 하기 위해, 폴더에 대한 사용자의 지식, 즉 단어의 유사성을 적용하였다.

이와 같은 목적에 의해 실제 데이터를 수집하여 실험한 결과 기존의 방법보다 본 연구를 통해 제시된 방법론의 추천 성능이 우수한 것으로 나타났으며, 특히 추천할 폴더의 수가 제한적일수록 단어의 유사성 적용에 대한 정확성 차이가 큰 것으로 나타나고 있다. 그러나 사용자의 메일의 양이 점점 누적이 된다면 기존의 TF-IDF 기반 프로파일이나 사용자의 지식이 부가된 프로파일이나 추천의 성능은 비슷해 질 것으로 보인다.

본 연구에 사용된 데이터는 실제 메일 데이터로써 자료 수집에 따른 한계로 인해 데이터의 수가 충분하지 않아 편중된 결과가 도출되었을 수도 있다. 또한 본 연구에서 실험에 사용된 사용자 지식 즉, 폴더에 대한 사용자의 지식은 사용자가 직접 정의하는 하였으나, 일반적으로 보통의 사용자들에게 단어의 유사성 정의는 매우 번거로운 작업이며, 불필요한 정보나 과다한 정보를 정리하는 것과 마찬가지로 추가적인 시간과 노력을 요하는 작업일 것이다. 따라서 향후 연구에서는 사용자의 지식을 자동으로 생성하고 반영할 수 있는 방안에 대해 고려해보고자 한다.

References

[1] Salton, G., and M. J. McGill (1983). *Introduction to modern information retrieval*. New York, McGraw Hill Book Company.

[2] Androutsopoulos, I., G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos (2000). "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach," *4th European Conference on Principles and Practice of Knowledge Discovery in Databases.*

[3] Balabanovic, M., and Y. Shoham (1997). "Fab:Content-Based, Collaborative Recommendation," *Communications of the ACM*, Vol. 40, pp. 66-72.

Belkin, N. J., and W. B. Croft(1992). "Information Filtering and Information Retrieval: Two Sides of the Same Coin?," *Communications of the ACM,* Vol. 35, pp. 29-38.

[4] Boone, G. (1998). "Concept Features in Re: Agent, an Intelligent Email Agent," *In Proceedings of the Second International Conferences on Autonomous Agents*

[5] Cohen, W. W. (1995). "Fast Effective Rule Induction," *Machine Learning: Proceedings of the Twelfth International Conference.*

[6] Cohen, W. W. (1996). "Learning Rules that Classify E-mail," *In Paper from the AAAI Spring Symposium on Machine Learning in Information Access.*

[7] Diao, Y., H. Lu, and D. Wu (2000). "A Comparative Study of Classification Based Personal E-mail Filtering," *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining.*

[8] Foltz, P. W., and S. T. Dumaiz (1992). "Personalized Information Delivery: An Analysis of Information Filtering Methods," *Communications of the ACM*, Vol. 35, pp. 51-60

[9] Kiritchenko, S., and S. Matwin (2001). "Email Classification with Co-Training," *IBM Centre for Advanced Studies Conference Proceedings*

[10] Kuflik, T., and P. Shoval (2000). "Generation of User Profiles for Information Filtering – Research Agenda," *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval.*

[11] Lee, J. K., J. K. Kim, S. H. Kim, and H. K. Park (2002). "An Intelligent Idea Categorizer for Electronic Meeting Systems," *Group Decision and Negotiation*, Vol. 11, pp. 363-378.

[12] Lewis, D. D., R. E. Schapire, J. P. Callan, and R. Papka (1996). "Training Algorithms for Linear Text Classifiers," *19th ACM International Conference on Research and Development in Information Retrieval.*

[13] Oard, D. W., and G.. Marchionini (1996). "A Conceptual Framework for Text Filtering," *Technical Report CS-TR3643*, University of Maryland.

[14] Pazzani, M. J. (2000). "Representation of Electronic Mail Filtering Profiles: A User Study," *Proceedings of the 5th international conference on Intelligent user interfaces.*

[15] Rennie, J. D. (2000). "ifile: An Application of Machine Learning to Email Filtering," *KDD-2000 Text Mining Workshop Boston.*

[16] Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work.*

[17] Romero, F.P., J. A. Olivas, P. Garces, and L. Jimenez (2003). "fzMail: A Fuzzy Tool for Organizing E-Mail," *Proceedings of the International Conference on Artificial Intelligence IC-AI'03.*

[18] Sahami, M., S. Dumais, D. Heckerman, and E. Horvitz (1998). "A Bayesian Approach to Filtering Junk E-mail," *In AAAI-98 Workshop on Learning for Text Categorization.*

[19] Segal, R. B., and J. O. Kephart (1999). " Mailcat: An Intelligent Assistant for Organizing E-mail," *In Proceedings of the Third International Conference on Autonomous Agents.*

[20] Segal, R. B., and J. O. Kephart (2000). "SwiftFile: An Intelligent Assistant for Organizing E-mail," *In Proceedings of the 2000 AAAI Spring Symposium on Adaptive User Interfaces.*

[21] Segal, R. B., and J. O. Kephart (2000). "Incremental Learning in SwiftFile," *In Proceedings of the Seventeenth International Conference on Machine Learning.*