

TPIPF로 계산된 사용자프로파일을 적용한 논문추천시스템에 대한 연구

A Study on Scientific Article Recommendation System with User Profile Applying TPIPF

장령령 (Lingling Zhang)*

장우권 (Woo Kwon Chang)**

초 록

오늘날 폭발적인 정보의 증가로 이용자들은 자신이 원하는 정보를 찾기 위해 엄청난 시간과 노력을 기울여야 한다. 이 문제를 해결하기 위하여 이용자의 정보요구를 분석하고 이용자에게 적합한 논문을 추천해주는 논문추천시스템이 등장하고 있다. 그러나 대부분의 논문추천시스템은 논문추천시스템의 핵심인 사용자 프로파일을 간과하고 있다. 따라서 이 연구는 논문추천시스템의 성능을 좌우하는 사용자 프로파일을 기존의 평균으로 계산하지 않고 새로운 TPIPF(Topic Proportion-Inverse Paper Frequency)로 계산하는 방법을 제안하였다. 제안된 방법과 기존의 방법을 모두 논문추천시스템에 적용하여 각각의 성능을 온라인 참고문헌 관리도구인 CiteULike에서 제공된 데이터 실험을 통하여 비교하였다. 그 결과 제안된 TPIPF 방법을 적용한 논문추천시스템의 성능이 더 높다는 것을 알 수 있었다.

ABSTRACT

Nowadays users spend more time and effort to find what they want because of information overload. To solve the problem, scientific article recommendation system analyse users' needs and recommend them proper articles. However, most of the scientific article recommendation systems neglected the core part, user profile. Therefore, in this paper, instead of mean which applied in user profile in previous studies, New TPIPF (Topic Proportion-Inverse Paper Frequency) was applied to scientific article recommendation system. Moreover, the accuracy of two scientific article recommendation systems with above different methods was compared with experiments of public dataset from online reference manager, CiteULike. As a result, the proposed scientific article recommendation system with TPIPF was proven to be better.

키워드: 논문추천시스템, 논문 프로파일, 사용자 프로파일, TPIPF, 콘텐츠기반 필터링, LDA
scientific article recommendation system, article profile, user profile, TPIPF,
content-based filtering, LDA

* 전남대학교 대학원 문헌정보학과 박사과정 수료(s06303018@gmail.com) (제1저자)

** 전남대학교 문헌정보학과 교수(wk1961@jnu.ac.kr) (교신저자)

■ 논문접수일자: 2016년 3월 5일 ■ 최초심사일자: 2016년 3월 5일 ■ 게재확정일자: 2016년 3월 22일
■ 정보관리학회지, 33(1), 317-336, 2016. [http://dx.doi.org/10.3743/KOSIM.2016.33.1.317]

1. 서론

정보통신기술과 웹 기술의 발전에 따라 많은 양의 학술정보가 지수함수적으로 증가하고 있다. 이러한 폭발적인 정보의 증가로 이용자들은 자신이 원하는 정보를 찾기 위해 많은 시간과 노력을 기울여야 한다. 이와 같은 문제를 해결하기 위해 다양한 방법들이 시도되었으며 가장 대표적인 방법으로 학술검색엔진을 사용하고 있다.

그러나 학술검색엔진은 질의어를 통하여 검색이 이루어지기 때문에 이용자는 자신이 원하는 정보가 무엇인지 정확히 알지 못하거나 서로 다른 이용자가 같은 질의어로 검색할 때 같은 결과가 나타나는 문제를 가지고 있다. 이에 이용자의 개인적인 정보요구를 보다 정확하게 분석하고 이를 기반으로 이용자에게 적합한 정보가 무엇인지를 예측하고 추천하는 시스템, 즉 논문추천시스템이 등장하고 있다.

주로 일부 전자도서관(여운동, 박현우, 권영일, 박영욱, 2010; Hwang & Chuang, 2004; Torres, McNee, Abel, Konstan, & Riedl, 2004; Vellino, 2010)과 이용자의 연구 활동 및 학술정보 공유와 협력을 지원하는 온라인 참고문헌 관리도구(Bogers & Bosch, 2008; Henning & Reichelt, 2008)에서 논문추천시스템을 적용하였다. 이들 논문추천시스템은 일반 추천시스템에서 많이 사용되는 콘텐츠기반 필터링(Hwang & Chuang, 2004), 협업 필터링(Bogers & Bosch, 2008; Henning & Reichelt, 2008; Vellino, 2010), 그리고 혼합 필터링 기법(여운동 외, 2010; Torres et al., 2004)을 사용해 왔다.

그러나 이들 전자도서관과 온라인 참고문헌

관리도구에서 추천된 논문이 특정논문과 연관성이 있다고 보기는 어렵다. 그 이유는 첫째, 추천시스템의 효율성이 뛰어난 협업 필터링은 음악, 영화 등의 분야에서 많이 사용되고 있어 반구조적인 특성을 가진 논문추천시스템에는 맞지 않다고 볼 수 있다. 둘째, 논문추천시스템에 콘텐츠기반 필터링을 적용하더라도 추천시스템의 핵심인 사용자프로파일에 대한 연구가 미흡한 실정이다. 셋째, 논문추천시스템의 주체는 논문이다 보니 반구조적인 특성으로 인해 논문 간의 의미적인 관계를 파악하는데 어려움이 있다.

따라서 이 연구에서는 논문의 의미적인 관계를 파악할 수 있을 뿐만 아니라 사용자프로파일도 중요시하는 TPIPF로 계산된 논문추천시스템을 제안하고자 한다.

2. 이론적 배경

2.1 기존의 논문추천시스템

논문추천시스템은 연구자의 흥미와 연구 분야를 반영하여 연구자에게 적절한 논문을 추천해주는 시스템으로 주로 콘텐츠기반 필터링, 협업필터링, 그리고 혼합필터링 기법을 적용하여 왔다.

콘텐츠기반 필터링은 정보검색이나 정보 필터링에 기반을 두고 있고 특정이용자가 선호하는 아이템 집합에 대해 내용을 분석하여 사용자의 특징을 생성하며 이와 비슷한 특징을 갖는 아이템을 추천하는 필터링기법이다(Lops, Gemmis, & Semeraro, 2011). 협업필터링은 특정 이용

자와 비슷한 취향을 가진 다른 이용자가 선호하는 다른 아이템을 특정 이용자에게 추천해주는 필터링기법이다(Sarwar, Karypis, Konstan, & Riedl, 2001). 혼합필터링은 콘텐츠기반 필터링과 협업필터링을 동시에 적용하는 기법으로 서로의 단점을 보완함으로써 추천시스템의 성능을 높일 수 있는 기법이다(Burke, 2002).

여기에서 콘텐츠기반 필터링은 뉴스, 문서 등과 같은 텍스트 분석에 우수한 성능을 보여주고 있다(Lops, Gemmis, & Semeraro, 2011). 따라서 콘텐츠기반 필터링을 적용한 논문추천시스템을 살펴보면 다음과 같다.

2.2 콘텐츠기반 필터링

콘텐츠기반 필터링과 관련된 논문은 콘텐츠기반 필터링의 프로세스에 따라 논문 프로파일, 사용자 프로파일 및 유사도 측정의 세 부분으로 나누어진다(Lops, Gemmis, & Semeraro, 2011; Sugiyama & Kan, 2010).

2.2.1 논문 프로파일

논문 프로파일은 논문을 표현하는 방법을 의미하며 보통 논문을 대표할 수 있는 키워드를 찾아내고 벡터 형식으로 각 키워드의 가중치를 부여하는 벡터공간모델(Vector Space Model)을 적용한다(Lops, Gemmis, & Semeraro, 2011). 여기서 논문을 대표할 수 있는 키워드는 plain words(논문에 있는 단어 그대로)(Huang, Kataria, Caragea, Mitra, Giles, & Rokach, 2012), WordNet 등과 같은 주제명(최호연, 신동욱, 최중민, 김정선, 2013), 온톨로지(Middleton, Roure, & Shadbolt, 2001), 개념(Pudhiyaveetil, Kodakateri,

Luong, & Eno, 2009), 주제(Jiang, Jia, Feng, & Zhao, 2012), 마인드 맵(Beel, Langer, Genzmehr, & Nürnberger, 2013), 태그(Choochaiwattana, 2010) 등의 형식이 있다.

여기에서 논문에 있는 단어는 동형의어와 이음동의어 등 단어 의미의 모호성으로 인해 논문의 의미적인 관계를 파악하기가 힘들다. 따라서 전문가에 의해 수동으로 작성한 온톨로지, 개념, WordNet 등과 같은 주제명을 이용하여 논문의 관계를 파악할 수 있다. 그러나 이러한 작업은 많은 시간과 노력이 필요하고 시스템의 자동화 성능을 저하시킬 수 있다. 이 외에 마인드 맵(mind map) 및 태그는 이용자에 의해 생성하기 때문에 일부 한정된 논문에서만 추출할 수 있다.

따라서 논문에서는 이산자료들에 대한 확률적 생성모형인 LDA(Latent Dirichlet Allocation)(David, Andrew, & Michael, 2003)를 적용하여 논문의 주제를 추출하고 추출된 주제를 논문의 키워드로 삼는 것은 적합하다. 이는 논문의 의미적인 관계를 파악할 수 있을 뿐만 아니라 전문가에 의존하지 않아도 되기 때문에 효율적이다. 이에 이 연구에서는 논문에 LDA를 적용하여 논문을 대표할 수 있는 주제를 추출하여 벡터 형식으로 표현한다.

2.2.2 사용자 프로파일

이용자 프로파일은 이용자가 어떤 논문을 좋아하는지를 알아보는 것이다. 이것은 과거의 경험(이력)을 통해서 이용자가 어떤 논문을 좋아하는지 그리고 좋아하는 논문의 특성이나 주제를 찾아내는 것이다(Lops, Gemmis, & Semeraro, 2011). 논문추천시스템에서 이용자가 논문을 좋

아하는 행위는 주로 이용자가 논문을 저장(Beel et al., 2013; Bogers & Bosch, 2008; Pudhiyaveetil et al., 2009)하거나 저자로서 논문을 쓰는(Chandrasekaran, Gauch, Lakkaraju, & Luong, 2008) 경우에 해당된다.

따라서 사용자프로파일은 이용자의 행위에 따라 이용자가 좋아하는 논문을 찾아내고, 이것을 논문 프로파일에 적용하여 벡터 형식으로 표현한 다음, 모든 논문의 벡터를 합하면 이용자가 좋아하는 논문의 특성이나 주제를 추출할 수 있다.

2.2.3 유사도 측정

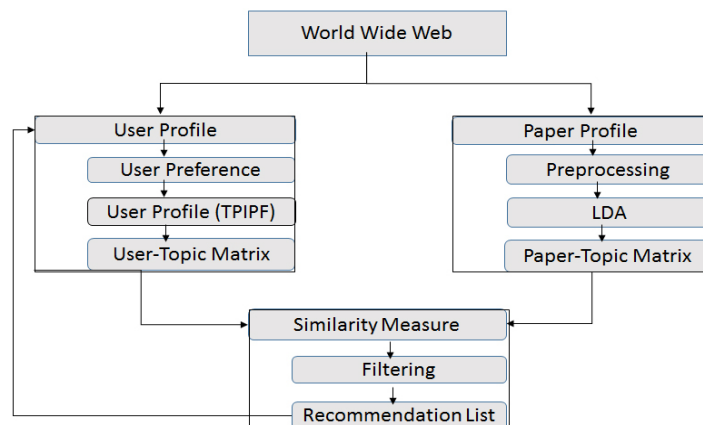
유사도 측정은 앞서 찾아낸 논문 프로파일과 이용자 프로파일의 유사도를 측정하고 이용자 프로파일에 없는 논문을 이용자에게 추천해주는 것이다. 대부분의 논문에서는 내적공간의 두 벡터간 각도의 코사인값을 이용하여 측정된 벡터간의 유사한 정도를 의미하는 코사인유사도(cosine similarity)(Choochaiwattana, 2010; Torres et al., 2004)를 사용했다. 또한 논문 프

로파일에서 키워드는 개념을 사용할 경우 TED (Tree Edit Distance)를 사용하는 논문도 있다(Chandrasekaran et al., 2008).

기존의 논문추천시스템을 살펴본 결과 콘텐츠 기반 필터링이 논문추천시스템의 특성에 맞는 것임을 알 수 있다. 또한 논문을 어떻게 표현할 수 있는지, 논문 간의 의미적인 관계를 어떻게 추출하는지는 논문 프로파일에 달려 있다. 이 외에 논문추천시스템이 검색엔진과 차별화된 것은 이용자 프로파일이고 이용자 프로파일이 논문추천시스템의 핵심이라고 해도 과언이 아니다. 따라서 이 연구에서는 논문추천시스템을 제안한다.

3. 제안방법

이 연구에서는 기존의 논문추천시스템의 현황을 살펴본 후 논문추천시스템의 특성에 맞는 콘텐츠기반 필터링을 채택하였다. 여기에서 논문의 의미적인 관계를 잘 파악할 수 있는 LDA



〈그림 1〉 TPIPF로 계산된 논문추천시스템의 구조

모델을 논문 프로파일에 적용하고 TFIDF의 원리를 응용한 TPIPF(Topic Proportion-Inverse Paper Frequency)를 사용자 프로파일에 적용한 새로운 논문추천시스템을 제안한다.

제안한 논문추천시스템의 구조는 위의 <그림 1>과 같이 세 부분으로 구성되어 있다.

TPIPF 기반 논문추천시스템은 우선 웹에서 이용자의 논문 선호도에 따라서 이용자가 좋아하는 논문을 수집하고 이들 논문과 관련된 제목, 초록, 논문 내용 등도 웹에서 스크랩한다. 그 다음으로 전체 논문은 말뭉치(Corpus)라고 하고 말뭉치에 LDA를 적용함으로써 논문-주제 행렬을 생성한다. 그리고 이용자 프로파일은 논문 프로파일의 논문-주제 행렬에 의하여 이용자-주제 행렬을 생성한다. 마지막으로 논문프로파일과 이용자 프로파일의 유사도를 계산하여 이용자에게 유사도가 높은 논문을 추천해준다.

3.1 논문프로파일

논문프로파일은 논문에서 주제와 관련된 내용에 LDA를 적용하여 논문-주제 행렬을 생성하기 위한 것이다. 따라서 논문에 LDA를 적용하기 전에 논문에서 주제를 추출하는 데에 방해되는 부분을 전처리(preprocessing) 과정을 거쳐 제거해야 한다.

따라서 이 연구에서는 다음과 같은 전처리 과정을 거쳤다.

첫째, 기호 및 숫자를 제거하였다. 둘째, 대문자를 소문자로 전환하였다. 셋째, 논문에 있는 불용어(stop word)는 포터 어간추출 프로그램(Porter Stemmer)(Willett, 2006)을 사용하여

제거하였다. 넷째, 스템밍(stemming)과 관련하여 두 가지 의견이 있다. 하나는 스템밍을 통하여 추천시스템의 재현율을 제고할 수 있다(박상진, 김윤현, 이지현, 2011; 최호연 외, 2013; Sugiyama & Kan, 2010)는 것이고, 다른 하나는 단어의 과거형, 진행형, 미래형을 보존해야 됴므로 스템밍을 권장하지 않은 의견(He, Pei, Kifer, Mitra, & Giles, 2010)이 있다. 이 연구에서 저자의 개인적인 경험으로는 스템밍을 할 경우에 단어의 원래의 뜻을 알아보기 힘들 때가 많아 실질적으로 토픽모델링에서 정확한 주제를 추출할 수 없고, 스템밍을 하지 않을 경우 추천시스템의 재현율이 낮아진다. 따라서 이 연구에서는 다음과 같은 방법을 채택하였다. 먼저 스템밍 과정을 거치고 스템밍한 단어는 원래 논문 중에서 제일 많이 나타난 단어를 기준으로 다시 한 번 변환시킨다. 이를 통하여 추출된 주제는 정확하고 시스템의 재현율도 높일 수 있다. 다섯째, 이상의 과정을 거쳐도 여전히 남은 단어가 많아서 다시 한 번 전체 논문 중에 출현빈도가 5번 이하인 단어를 제거(Griffiths & Steyvers, 2004)하거나 전체 단어의 TFIDF 값을 구하고 낮은 TFIDF 값을 가진 단어를 제거하는 방법(Blei & Lafferty, 2009)이 있다. 본 연구에서는 TFIDF 값을 구한 단어 중 출현빈도가 많은 단어도 제거되어 논문 간의 유사도를 저하시켰다는 판단에 따라 첫 번째 방법을 채택하였다.

위와 같은 전처리 과정을 거친 말뭉치에 LDA를 적용하였다. 그 이유는 첫째, 이 토픽모델링을 통하여 각 주제가 논문에 차지하는 비율을 계산할 수 있다. 둘째, LDA를 통하여 논문은 논문에 있는 단어 그대로(plain words) 표현되는 것이 아니고 주제로 표현되기 때문에 단어

의미의 모호성을 제거하고 논문의 의미적인 관계를 파악할 수 있다. 셋째, 논문은 단어의 고차원에서 주제의 저차원으로 축소시키면서 컴퓨터 메모리를 절약하는 것은 물론이고 계산 속도도 빨라지고 추천시스템의 효율성을 제고할 수 있다.

따라서 논문 프로파일을 행렬로 표현하면 <그림 2>와 같다.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

<그림 2> 논문 프로파일의 논문-주제 행렬

이 논문-주제 행렬은 행으로 보면 이용자가 각 주제에 대한 흥미도를 나타내는 수치이다.

3.2 이용자 프로파일

3.2.1 기존의 평균으로 계산된 이용자 프로파일

기존의 논문추천시스템에서 이용자 프로파일은 논문 프로파일에 의하여 생성된 논문-주제 행렬을 기반으로 이용자가 좋아하는 논문을 합해서 이용자가 좋아하는 논문의 편수를 나누고, 즉 이용자가 좋아하는 논문의 논문-주제 행렬의 평균을 구하는 것이다(Jiang et al., 2012). 또한 이용자는 주제에 대한 흥미도를 벡터 형식으로 표현하고 모든 벡터를 모아서 주제-이용자 행렬을 생성한다.

이용자의 주제에 대한 흥미도를 공식으로 나타내면 다음과 같다.

$$T_{u(m)} = \frac{1}{|J|} \sum_{i \in J} \vec{a}_i$$

J 는 이용자가 좋아하는 논문의 수이며 \vec{a}_i 는 이용자가 좋아하는 논문의 주제 벡터이다.

3.2.2 TPIPF로 계산된 이용자 프로파일

TPIPF는 이용자가 좋아하는 주제 벡터를 평균으로 구하는 것이 아니고 TFIDF와 같은 원리로 이용자가 좋아하는 논문의 주제에 가중치를 매기는 방법이다. 전체 주제를 보면 많은 논문에서는 나타나지만 비중이 더 크지 않은 주제의 가중치는 적게 하고 많은 논문에서는 나타나지 않지만 하나의 논문에 큰 비중이 차지하는 것은 주제의 가중치를 높게 한다. 이와 같은 과정과 방법을 공식으로 나타내면 다음과 같다.

$$T_{u(TPIPF)} = \frac{1}{|J|} \sum_{i \in J} (\vec{a}_i) * \log(1 + \frac{|J|}{|j|})$$

이 공식에서 J 는 위의 공식과 같이 이용자가 좋아하는 논문의 수, $|J|$ 는 이용자가 좋아하는 논문 중 주제가 0이 아닌 논문의 수이다.

<표 1>은 평균 및 TPIPF 두 가지 방법으로 계산된 이용자 프로파일의 차이를 보여주고 있다. 예를 들면, 이용자가 좋아하는 논문은 6편이 있고 이 6편의 논문의 주제가 6개가 있다고 가정한다.

<표 1>을 보면 주제1은 논문1에서만 나타나고 그러나 차지하는 비율이 0.7이며 논문1의 전체 주제 중에 차지하는 비율이 높아서 이용자가 주제1을 좋아한다고 볼 수 있다. 주제2도 주

〈표 1〉 평균과 TPIPF로 계산된 이용자 프로파일

	주제1	주제2	주제3	주제4	주제5	주제6
논문1	0.7	0	0.2	0.05	0.05	0
논문2	0	0	0.5	0.3	0.2	0
논문3	0	0	0.3	0.2	0.2	0.3
논문4	0	0.6	0.3	0	0.1	0
논문5	0	0	0.7	0	0.1	0.2
논문6	0	0.1	0.4	0.2	0	0.3
평균	0.116667	0.116667	0.4	0.125	0.108333	0.133333
TFIDF	0.22702	0.161734	0.277259	0.114536	0.085416	0.146482
min-max	0.73814	0.397816	1	0.151792	0	0.31831
z-score	0.814525	-0.09793	1.516614	-0.75756	-1.16454	-0.3111

제1과 마찬가지로 2개의 논문에서만 나타나지만 논문4의 전체 주제 중에 차지하는 비율이 높아서 이용자가 주제2를 좋아한다고 생각할 수 있다. 반면에 주제4는 네 편의 논문에 나타나지만 각 논문에 주제가 차지하는 비율이 낮아서 이용자는 주제4를 좋아한다고 보기는 힘들다.

그러나 평균으로 계산된 이용자 프로파일을 보면 이용자가 좋아하는 주제는 주제3, 주제6, 주제4, 주제1=주제2, 주제5의 순서로 나타났다. 주제1과 주제2는 주제4 뒤에 나와 있는 것으로 나타나 불합리하다고 볼 수 있다. 반면에 TPIPF로 계산된 이용자 프로파일을 보면 주제3, 주제1, 주제2, 주제6, 주제4, 주제5의 순서로 나타났다. 주제3은 모든 논문에서 나타났고 각 논문에서 차지하는 비율도 낮지 않아 당연히 이용자가 제일 좋아하는 주제라고 생각할 수 있다. 그러나 평균으로 계산된 이용자 프로파일과 차이가 있는 부분은 주제1과 주제2의 순서에 있고 TPIPF로 계산된 이용자 프로파일은 주제1과 주제2의 순서를 더 앞당겼다. 그 결과 TPIPF로 계산된 이용자프로파일은 이용자의 주요 관심분야인 주제3을 간과하지 않

면서 이용자가 좋아하는 주제1과 주제2도 낮추지 않은 것은 장점이다.

마지막으로 TPIPF로 계산된 이용자의 주제 흥미도의 합은 1이 아니기 때문에 정규화해야 한다. 일반적으로 정규화 방법은 두 가지가 있는데 하나는 최소-최대(min-max)이고 하나는 z 점수(z-score)이다. 최소-최대는 데이터의 범위를 [0,1]로 좁힐 수 있다. 그 공식은 다음과 같다.

$$t_{i(\min-\max)} = \frac{t_i - \min_{t_i}}{\max_{t_i} - \min_{t_i}}$$

z 점수는 데이터의 평균 및 표준편차를 통하여 정규화 한다. 그 공식은 다음과 같다.

$$t_{i(z-score)} = \frac{t_i - \mu_{t_i}}{\sigma_{t_i}}$$

그러나 〈표 1〉에서 나타난 바와 같이 z 점수의 결과로 마이너스 값이 나타났는데 이용자가 주제에 대한 흥미도가 마이너스 값이 나타날 수

없기 때문에 데이터의 범위를 [0,1] 사이로 좁히는 최소-최대 정규화를 택하였다.

3.3 유사도 측정

유사도 측정은 논문-주제 행렬로 된 논문 프로파일과 이용자-주제 행렬로 된 이용자 프로파일의 유사도를 측정하고 이용자 프로파일에 있는 논문을 제거하여 유사도에 값을 높은 순서로 배열하여 이용자에게 일정 수의 논문을 추천해주는 것이다. 대부분의 논문에서 코사인 유사도를 계산하기 때문에 이 논문에서도 코사인 유사도를 채택하면 그 공식은 다음과 같다.

$$\text{sim}(P_{user}, P_{paper}) = \frac{P_{user} \cdot P_{paper}}{|P_{user}| \cdot |P_{paper}|}$$

이 연구에서는 연구자가 제안한 LDA를 적용한 논문프로파일, TPIPF로 계산된 이용자 프로파일, 유사도 측정의 세 부분으로 구성된 논문추천시스템에 대하여 상세하게 기술하였다. 다음 장에서는 제안된 논문추천시스템의 정확률은 어떻게 실험을 통하여 검증할 것인지를 알아보고자 한다.

4. 실험 및 평가

4.1 논문추천시스템의 평가방법

논문추천시스템의 정확률을 측정하는 방법은 주로 오프라인 실험, 이용자연구, 온라인 실

험의 세 가지(Shani & Gunawardana, 2011)가 있다. 이 연구에서는 논문추천시스템의 프로토타입(prototype)을 개발하는데 있어서 실제 이용자와의 상호작용이 필요 없으면서 효율성이 제일 뛰어난 오프라인 실험을 채택하기로 한다.

4.2 평가지표

이 연구에서는 오프라인 실험에서 정확률과 관련이 있으며 널리 쓰이는 세 가지 지표를 사용하기로 한다.

4.2.1 precision@K

정확률(precision@K)(Bogers & Bosch, 2008)은 이용자에게 추천해준 앞 K개의 논문이 이용자가 실제로 좋아하는 논문 리스트에 차지하는 비율이며 그 공식은 다음과 같다.

$$P@K = \frac{N_r}{K}$$

여기에서 K는 이용자에게 추천해주는 논문의 수이며 보통 5나 10로 정하며 N_r 는 추천된 논문 중에 이용자가 실질적으로 좋아하는 논문의 수이다.

4.2.2 MAP(Mean Average Precision)

추천된 논문이 이용자가 선호하는 논문에 차지하는 비율이 중요하지만 추천된 논문 리스트의 순서도 중요하다. MAP와 MRR(Bogers & Bosch, 2008)는 추천논문 리스트의 순서를 평가하는 지표이다. MAP의 계산방법은 다음과 같다.

$$MAP = \frac{1}{|U|} \sum_{j=1}^{|U|} \frac{1}{m_j} \sum_{k=1}^N P(R_{jk})$$

$|U|$ 는 이용자의 수이며, m_j 는 이용자 j 가 선호하는 논문의 수, $P(R_{jk})$ 는 추천된 논문이 논문 k 까지 정확률이다.

4.2.3 MRR(Mean Reciprocal Rank)

MRR(Bogers & Bosch, 2008) 지표는 MAP와 마찬가지로 추천논문 리스트의 순서를 평가하는 지표이고 계산방법은 다음과 같다.

$$MRR = \frac{1}{|U|} \sum_{j=1}^{|U|} \frac{1}{rank_{F_j}}$$

$rank_{F_j}$ 는 이용자 j 가 선호하는 논문의 위치이다.

4.3 실험방법

4.3.1 데이터 수집

TPIPF로 계산된 사용자 프로파일은 논문추천시스템의 성능을 알아보기 위해 CiteULike 사이트에서 1년 동안(2015.1.1-2015.12.31)의 데이터를 수집하여 검정하였다. CiteULike는 이용자가 온라인으로 참고문헌을 관리하는 사이트이며 이용자가 좋아하는 논문을 자기의 라이브러리(Library)에 담고 논문에 대한 태그도 추가할 수 있다. 또한 이 사이트는 지금까지 이용자가 자신의 라이브러리에 담긴 논문, 이용자가 가입한 그룹, 논문에 대한 정보의 세 가지 데이터를 날마다 공개한다(CiteULike, 2016). 여기에서는 이용자가 자신의 라이브러리에 담긴 논문만을 이용한다. 그러나 이 데이터에서는 논

문의 초록 및 제목 내용이 없어서 이 사이트의 R 프로그램을 통하여 JSON형식으로 다운 받았다.

또한 하나의 논문은 저자, 출판연도, 수록된 저널 등 메타데이터 및 제목, 초록, 키워드, 참고문헌, 본문 등 논문 내용의 두 부분으로 구성 된다고 할 수 있다. 그러나 이 연구에서는 논문에 토픽모델링을 적용하기 때문에 메타데이터 및 참고문헌을 제거하여 논문의 내용만 분석하기로 한다. 그리고 키워드는 보통 초록에 포함되어 있어서 키워드를 제거하였으며 본문은 내용이 많고 논문추천시스템에 잡음만 초래하기 때문에 본문도 토픽모델링에 적용하지 않는다. 이외에 CiteULike에서 이용자가 논문을 읽고 나서 이 논문에 대하여 자기의 의견 및 노트를 기록하기 위하여 태그를 논문에 추가한다(CiteULike, 2016). 이 태그는 논문의 주제가 될 확률이 높아서 태그도 논문을 표현할 수 있다. 따라서 논문의 제목, 초록, 그리고 태그를 이용하여 토픽모델링을 하고 논문의 주제를 찾아내기로 한다.

논문추천시스템의 성능을 검정하기 위하여 오프라인 실험에서 널리 쓰이는 교차검증(Cross-Validation)(James, 2014)을 택하였다. 이 방법은 이용자가 자기의 라이브러리에 저장한 논문을 훈련집합(Training Set)과 테스트집합(Test Set)으로 나누어 훈련집합에 변수를 최적화하고 모델을 학습하고 테스트집합에 논문추천시스템의 성능을 검정한다. 그리고 논문추천시스템의 성능을 계산하기 위하여 논문을 한 번만 저장한 이용자를 제거하고 전체 논문 중에 한 번만 나타난 논문은 잡음만 초래하기 때문에 이들을 제거하였다. 이와 같이 정리한 데이터는 <표 2>와 같다.

〈표 2〉 CiteULike의 데이터

이용자	논문	이용자-논문 쌍	한 사람 당 논문 수
208	1,633	1,663	7.85

4.3.2 실험환경

본 실험에서는 Windows 8 운영체제에 메모리 16GB, CPU Intel(R) Core(TM)인 컴퓨터를 활용하였다. 실험 데이터의 수집은 R 소프트웨어의 XML, httr, RCurl 등 패키지를 이용하였다. 또한 논문프로파일에 tm, topicmodels 등 패키지를 사용하였으며, 이용자 프로파일 및 유사도 측정은 R 소프트웨어를 이용하여 프로그래밍을 하였다. 마지막으로 R의 패키지인 래티스(lattice)를 이용하여 데이터를 가시화하였다(Sarkar, 2008).

4.3.3 실험설계

본 연구에서는 제안한 TPIPF로 구현된 논문추천시스템에 대한 검증을 위하여 두 가지 분야의 실험을 실시하였다. 첫째, 제안된 추천방법에서 관련된 요소 변경을 통하여 최적의 요소를 추출하기 위한 실험을 실시하였다. 둘째, TPIPF와 평균으로 계산된 이용자 프로파일에 대하여 세 가지 평가지표를 기준으로 논문추천시스템의 성능평가를 실시하였다.

- LDA를 실시하기 위하여 먼저 주제의 개수 k 및 추리방법을 어떻게 정할 것인지를 실험하였다.

이상의 실험결과를 기준으로 다음과 같이 이 연구에서 제안한 TPIPF로 구현된 논문추천시스템과 평균으로 구현된 시스템의 성능에 대한 비교평가를 위한 실험을 수행하였다.

- 평균과 TPIPF로 계산된 이용자 프로파일이 세 가지의 평가지표를 기준으로 논문추천시스템의 성능에 어떤 영향을 미칠 것인지를 실험하였다.
- 추천된 논문의 개수가 다르다면 평균과 TPIPF가 논문추천시스템의 성능에 어떤 영향을 끼칠 것인지를 실험하였다.
- 이용자 프로파일의 크기가 다르다면 TPIPF로 구현된 논문추천시스템의 성능에 어떤 영향을 미칠 것인지를 실험하였다.

4.4 실험결과 및 분석

4.4.1 LDA 주제의 수 및 추리방법

LDA를 논문에 적용하기 전에 주제의 수를 먼저 정해야 하는데 정해진 규칙이 없어서 보통 여러 가지의 주제의 수를 모델에 실험해보고 어느 주제의 수가 모델을 최적화할 수 있는지를 통하여 주제의 수를 정한다(Grün & Hornik, 2011). 이때 모델의 성능을 평가하는 척도는 Perplexity(확률분포의 혼란)이고 공식은 다음과 같다. 여기에 $n^{(jd)}$ 는 j 번째 단어가 d 번째 논문에 나타난 빈도이고 Perplexity가 적을수록 추천시스템의 성능이 더 좋다고 해석할 수 있다. 그러나 Perplexity를 계산하는 데에 $\log(p(w))$ 를 먼저 알아야 되고 $\log(p(w))$ 는 실질적으로 계산할 수 없고 추론을 통하여 최대한 이와 가까이에 근접하기만 할 수 있다. 따라서 이 값을 추리하는 방법은 VEM, VEM_fixed, Gibbs의 세

가지가 있다.

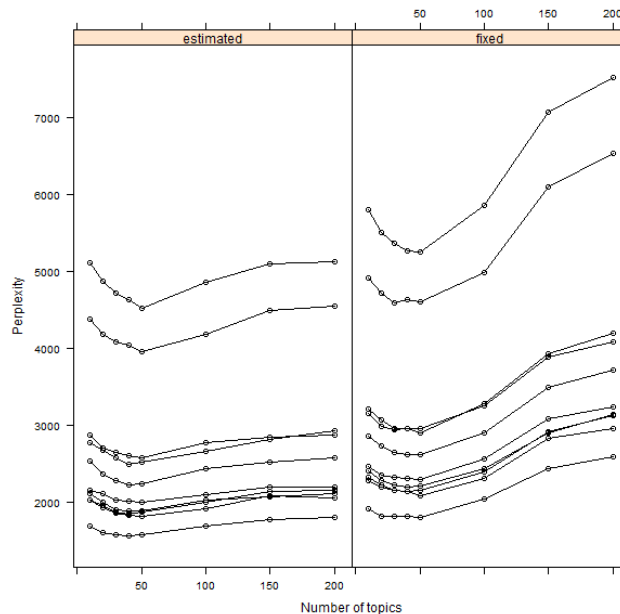
$$Perplexity(w) = \exp \left\{ - \frac{\log(p(w))}{\sum_{d=1}^D \sum_{j=1}^V n^{(jd)}} \right\}$$

- VEM: 이 방법은 LDA를 소개하는 논문에서 쓰이는 방법이고 관련된 파라미터 α , β 가 고정적이지 않고 모델의 공산(公算)에 따라서 변화하고 공산이 최대의 수치에 정해진다(Blei et al., 2003).
- VEM_fixed: 이 방법은 관련된 파라미터 α , β 는 각각 $50/k$, 0.01로 정한다(Griffiths & Steyvers, 2004).
- Gibbs: 이 방법은 샘플하는 방법으로 Markov-Chain Monte Carlo(MCMC)에 속하고 간편하고 계산량이 상대적으로 적다(Wallach, 2008).

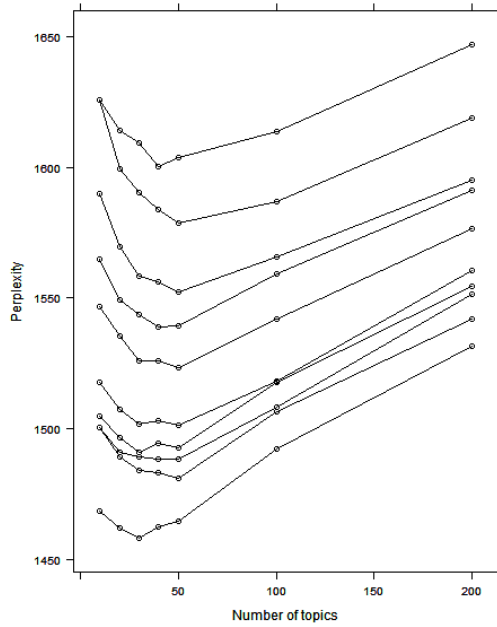
〈그림 3〉과 〈그림 4〉는 VEM, VEM_fixed, Gibbs 방법에 주제수와 Perplexity의 관계를 나타내는 것이다. 〈그림 3〉을 보면 주제의 수는 40-50일 때 Perplexity가 제일 낮은 것으로 알 수 있고 〈그림 4〉를 보면 30 혹은 50이 적합하다고 볼 수 있어서 LDA의 주제의 수가 50으로 정한다.

4.4.2 평균과 TPIPF로 구현된 논문추천시스템의 성능에 대한 비교

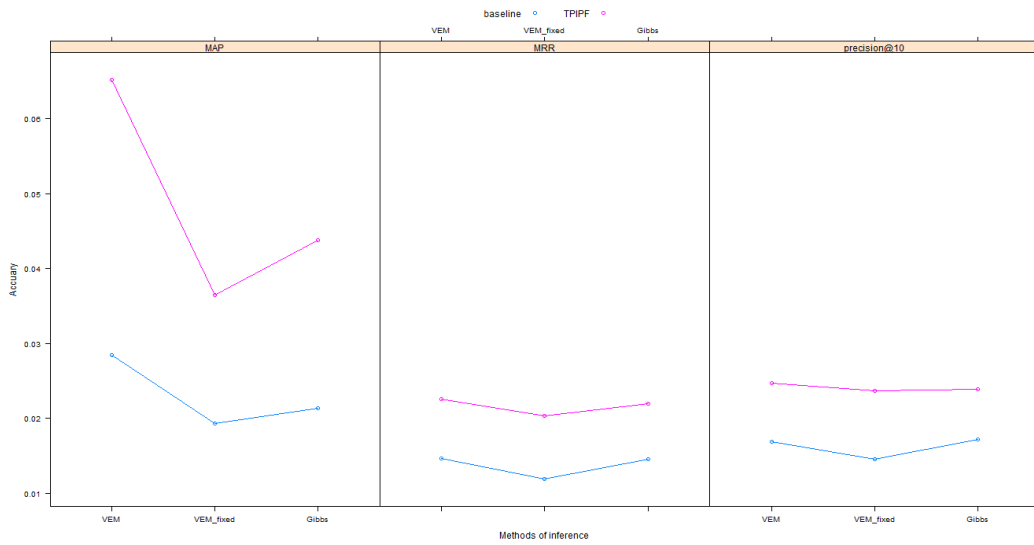
주제의 수를 정한 다음에 평균과 TPIPF로 계산된 이용자 프로파일은 논문추천시스템의 성능에 대한 영향을 비교하기 위하여 VEM, VEM_fixed, Gibbs의 세 가지 추리방법에 따라 precision@10, MAP, MRR 세 가지 평가지표를 기준으로 실험하였다.



〈그림 3〉 VEM 및 VEM_fixed와 Perplexity의 관계



〈그림 4〉 Gibbs와 Perplexity의 관계



〈그림 5〉 평균과 TPIPF로 구현된 논문추천시스템의 성능 비교

〈그림 5〉는 각각 VEM, VEM_fixed, Gibbs의 추리방법에 평균과 TPIPF로 계산된 이용자 프로파일의 논문추천시스템 성능에 대한 것

이다. 여기에서 〈그림 5〉를 보면 TPIPF로 계산된 이용자 프로파일로 구현된 논문추천시스템의 성능은 모두 세 가지 추리방법에서, 모두

세 가지 평가척도에서, 평균으로 구현된 논문 추천시스템의 성능보다 좋다고 나타났다.

또한 TPIPF로 구현된 논문추천시스템의 성능이 세 가지 평가지표와 세 가지 처리방법에서 평균으로 구현된 논문추천시스템의 성능보다 더 높다. 이것이 통계적으로 의미가 있는지를 t-test 한 결과는 <표 3>과 같다. p값이 0.001779로 나타났다. 이와 같이 TPIPF로 구현된 논문추천시스템의 성능이 평균으로 구현된 논문추천시스템의 성능보다 더 좋은 것은 통계적으로도 의미가 있다고 볼 수 있다.

마지막으로 세 가지 처리방법인 VEM, VEM_fixed, Gibbs가 논문추천시스템의 성능에 대한 영향이 다른지를 검정하기 위하여 ANOVA test 를 한 결과는 <표 4>와 같다.

<표 4>를 보면 세 가지 처리방법은 논문추천시스템의 성능에 대해 그 영향이 통계적으로 다르다고 보기 어렵다(p=0.607). 그럼에도 불구하고 그림에서 세 가지 평가지표에서 VEM이 VEM_fixed, Gibbs보다 약간 높은 것으로 나타났다. 따라서 다음으로 주제의 수가 50이고 처리방법인 VEM일 때 추천된 논문의 수에 따

라 평균과 TPIPF로 구현된 논문추천시스템의 성능은 어떻게 다른지를 실험하였다.

4.4.3 추천된 논문의 수

추천된 논문의 수는 논문추천시스템의 성능에 대해 영향을 미치고 논문의 수가 많을수록 논문추천시스템의 성능도 높아지는 것으로 생각할 수 있다. 먼저 사용자 프로파일의 크기에 대하여 계산해보았다. <표 5>는 사용자 프로파일의 크기에 대한 빈도와 누적비율이다. <표 5>에서 볼 수 있듯이 추천된 논문의 수를 25로 정하면 누적비율이 거의 95% 이상임으로 추천된 논문의 수를 10, 15, 20, 25로 변경하여 세 가지 평가지표를 기준으로 평균 및 TPIPF로 구현된 논문추천시스템에 대한 성능을 비교하였다.

<그림 6>은 추천된 논문의 수에 따라 평균과 TPIPF로 구현된 논문추천시스템의 성능의 변화를 보여주고 있다. 여기에서 TPIPF로 구현된 논문추천시스템의 성능이 세 가지 평가지표에서 평균으로 구현된 것보다 좋다고 볼 수 있다. 이것이 통계적으로도 의미가 있는지를 검증하기 위하여 t_test를 하였다. 그 결과는 <표 6>과 같

<표 3> 세 가지 계산방법에 따라 평균 및 TPIPF로 구현된 논문추천시스템의 성능에 대한 차이

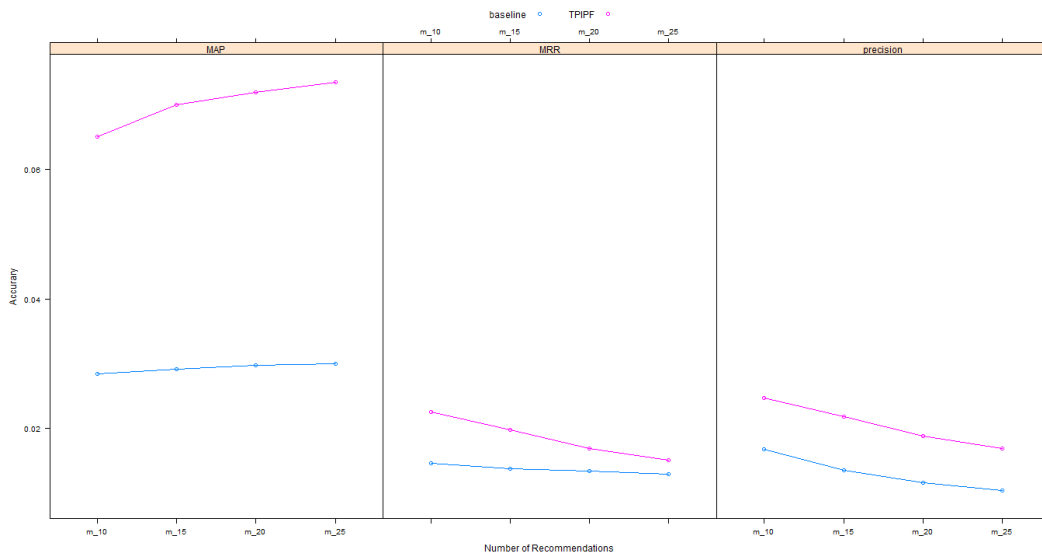
Paired t-test	평균	표준편차	t	df	p-value
baseline	0.01761925	0.00494434	4.075	8	0.001779
TPIPF	0.03138211	0.01487155			

<표 4> 세 가지 계산방법이 논문추천시스템의 성능에 대한 영향

처리방법	평균	표준편차	F value	p-value
VEM	0.02870143	0.01855998	0.517	0.607
VEM_fixed	0.02102423	0.008660126		
Gibbs	0.02377637	0.01037647		

〈표 5〉 이용자 프로파일의 크기

이용자 프로파일의 크기	빈도	누적비율	이용자 프로파일의 크기	빈도	누적비율
2	58	27.885%	19	2	90.385%
3	38	46.154%	20	1	90.865%
4	24	57.692%	21	1	91.346%
5	9	62.019%	22	3	92.788%
6	6	64.904%	23	1	93.269%
7	9	69.231%	24	2	94.231%
8	4	71.154%	25	1	94.712%
9	6	74.038%	26	1	95.192%
10	7	77.404%	27	2	96.154%
11	4	79.327%	29	2	97.115%
12	3	80.769%	34	1	97.596%
13	2	81.731%	36	1	98.077%
14	3	83.173%	40	1	98.558%
15	7	86.538%	48	1	99.038%
16	2	87.500%	66	1	99.519%
17	1	87.981%	78	1	100.000%
18	3	89.423%			



〈그림 6〉 추천된 논문의 수와 논문추천시스템의 성능 간의 관계

〈표 6〉 추천된 논문의 수에 따라 평균 및 TPIPF로 구현된 논문추천시스템의 성능에 대한 차이

Paired t-test	평균	표준편차	t	df	p-value
baseline	0.01870594	0.008005371	3.5701	11	0.002197
TPIPF	0.03642524	0.025090420			

다. <표 6>에서 알 수 있듯이 p값이 0.002197로 나타나 TPIPF로 구현된 논문추천시스템의 성능이 평균으로 구현된 것보다 좋다. 이것은 통계적으로도 의미가 있다.

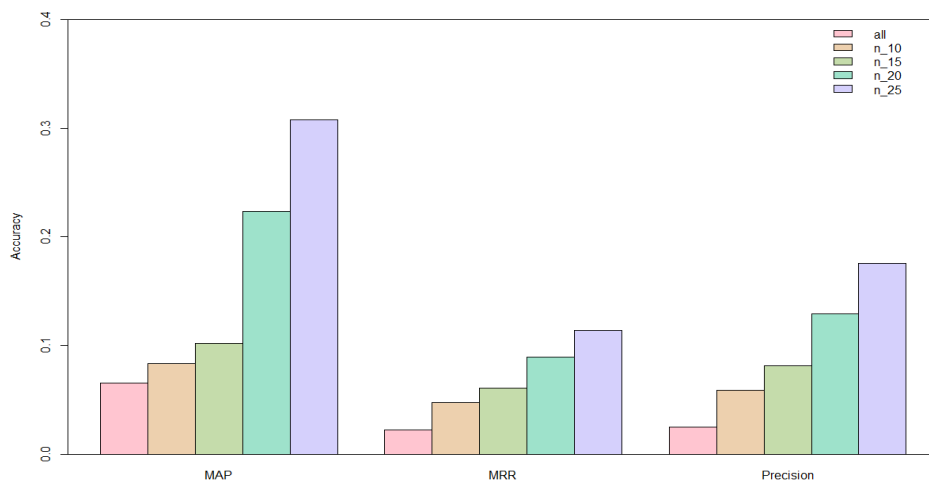
또한 평균과 TPIPF로 구현된 논문추천시스템의 성능은 추천된 논문의 수가 많을수록 MAP의 평가지표에서는 높아지고 MRR 및 정확률(precision)의 평가지표에서는 낮아지는 경향을 보이고 있다. 이것은 사용자 프로파일의 크기가 대부분 10개 이하에서 나타나고 있다고 볼 수 있다. 그러나 ANOVA test를 한 결과는 <표 7>과 같이 통계적으로 추천된 논문의 수에 따라 논문추천시스템의 성능이 다르다고 보기가 어렵다(p=0.998).

4.4.4 사용자프로파일의 크기

사용자 프로파일의 크기가 제안한 TPIPF로 구현된 논문추천시스템의 성능에 어떤 영향을 미칠 것인지를 검증하였다. 위의 <표 5>에서와 같이 사용자 프로파일의 크기는 전체 사용자, 10개 이상의 논문을 가진 사용자, 15개 이상의 논문을 가진 사용자, 20개 이상의 논문을 가진 사용자, 25개 이상의 논문을 가진 사용자로 나누어 제안한 논문추천시스템의 성능에 대해 그 영향을 실험하였다. 그 결과는 <그림 7>과 같다. <그림 7>을 보면 사용자 프로파일의 크기가 클수록 논문추천시스템의 성능은 좋아지는 것으로 나타났다.

<표 7> 추천된 논문의 수가 논문추천시스템의 성능에 대한 영향

추천된 논문의 수	평균	표준편차	F value	p-value
m_10	0.02870143	0.01855998	0.012	0.998
m_15	0.0280265	0.02135044		
m_20	0.02706142	0.02286878		
m_25	0.02647301	0.02399762		



<그림 7> 사용자 프로파일의 크기가 제안한 논문추천시스템의 성능에 대한 영향

〈표 8〉 이용자 프로파일의 크기에 따라 논문추천시스템의 성능에 대한 영향

이용자프로파일의 크기	평균	표준편차	F value	p-value	사후검증
all	0.02870143	0.01855998	10.12	5.19E-05	n_20>all, n_25>n_10, n_15, all (TukeyHSD)
n_10	0.06591843	0.02223957			
n_15	0.08744836	0.0292561			
n_20	0.1346146	0.05334156			
n_25	0.1887713	0.08326324			

이것은 이용자가 가지는 논문의 수가 많을수록 이용자에 대한 정보를 더 많이 알 수 있고 이용자에게 더 맞는 논문을 추천해줄 있을 것으로 해석할 수 있다. 또한 이용자 프로파일의 크기에 따라 논문추천시스템의 성능이 다르고 통계적으로 의미가 있는지를 알아보기 위해 ANOVA test를 실시하였다. 그 결과는 〈표 8〉과 같다.

〈표 8〉에서 알 수 있듯이 이용자 프로파일의 크기에 따라 논문추천시스템의 성능이 차이가 있다. 또한 사후 검증한 결과 25개 이상의 논문을 가진 이용자로 구현된 논문추천시스템의 성능이 10개, 15개 이상의 논문을 가진 이용자, 그리고 전체이용자로 구성된 논문추천시스템의 성능보다 더 나은 것으로 나타났다. 또한 20개 이상의 논문을 가진 이용자로 구현된 논문추천시스템의 성능은 전체 이용자로 구성된 논문추천시스템의 성능보다 더 나음을 알 수 있다.

5. 결론

이 논문은 논문추천시스템의 핵심인 이용자 프로파일에 TFIDF와 같은 원리를 적용한 TPIPF를 제안하고 논문추천시스템의 성능을 향상시키고자 하는 것이다. 조사와 실험결과는 다음과 같다.

첫째, 기존에 평균으로 구현된 논문추천시스템의 성능과 비교하기 위하여 먼저 논문프로파일에 적용할 LDA에 필요한 선행조건인 주제의 개수 및 추리방법을 LDA에 최적화할 수 있도록 하였다.

둘째, 평균과 TPIPF로 구현된 논문추천시스템의 성능을 정확률, MAP, MRR의 세 가지 평가지표를 기준으로 비교하였다. 그 결과 TPIPF로 구현된 논문추천시스템의 성능이 기존에 평균으로 구현된 논문추천시스템의 성능보다 세 가지 평가지표에서 모두 나은 것으로 나타났다. 또한 이를 통계적으로 입증하기 위하여 t-test한 결과도 역시 이와 같다.

셋째, 추천된 논문의 개수가 많아지면 논문추천시스템의 성능은 오히려 낮아지는 경향을 보인다. 이것은 대부분 이용자 프로파일의 크기가 10개 이하에서 그 원인을 찾을 수 있다. 그러나 추천된 논문의 개수가 달라도 TPIPF로 구현된 논문추천시스템의 성능이 평균으로 구현된 논문추천시스템의 성능보다 좋다는 것을 통계적으로 검정할 수 있다.

향후 연구에서는 전처리 과정에서 세 가지 스테밍 방법 및 두 가지 단어를 제거하는 방법에서 논문추천시스템의 성능에 대한 영향을 검정할 필요가 있다. 또한 LDA 모델의 주제간의 관계를 나타낼 수 없는 단점을 보완한 CTM(Correlated

Topic Models)(Blei & Lafferty, 2007)이 있다. 이 모델을 논문추천시스템에 적용하여 본 연구에서 제안한 TPIPF로 구현된 논문추천시스템의 성능과 비교하는 것을 기대한다.

참 고 문 헌

- 박상진, 김윤현, 이지현 (2011). 인용논문 분석을 통한 학술 문서 추천 시스템. 한국정보처리학회 춘계학술발표대회 논문집, 18(1), 279-282.
- 여운동, 박현우, 권영일, 박영욱 (2010). 연구논문 추천시스템의 전자도서관 적용방안. 한국콘텐츠학회 논문지, 10(11), 10-19. <http://dx.doi.org/10.5392/jkca.2010.10.11.010>
- 최호연, 신동욱, 최중민, 김정선 (2013). 전자 도서관 도메인에서 의미적 관계를 이용한 개인화된 논문 추천 시스템. 정보과학회논문지: 소프트웨어 및 응용, 40(3), 164-175.
- Beel, J., Langer, S., Genzmehr, M., & Nürnberger, A. (2013). Introducing docear's research paper recommender system. Proceedings of the 13th ACM/IEEECS Joint Conference on Digital Libraries, 459-460. <http://dx.doi.org/10.1145/2467696.2467786>
- Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science. The Annals of Applied Statistics, 1(1), 17-35. <http://dx.doi.org/10.1214/07-aos114>
- Blei, D. M., & Lafferty, J. D. (2009). Topic models. In A. Srivastava & M. Sahami (eds.), Text mining: classification, clustering, and applications. Chapman & Hall/CRC Press.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. Journal of Machine Learning Research, 3, 993-1022.
- Bogers, T., & van den Bosch, A. (2008). Recommending scientific articles using CiteULike. Proceedings of the 2008 ACM Conference on Recommender Systems, 287-290. <http://dx.doi.org/10.1145/1454008.1454053>
- Burke, R. (2002). Hybrid recommender systems: survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370. <http://dx.doi.org/10.1109/dictap.2012.6215409>
- Chandrasekaran, K., Gauch, S., Lakkaraju, P., & Luong, H. (2008). Concept-based document recommendations for Citeseer authors. Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 83-92. http://dx.doi.org/10.1007/978-3-540-70987-9_11
- Choochaiwattana, W. (2010). Usage of tagging for research paper recommendation. Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 439-442. <http://dx.doi.org/10.1109/icacte.2010.5579321>

- CiteULike (2016). dataset. Retrieved from <http://www.citeulike.org/faq/data.adp>.
- Ferrara, F., Pudota, N., & Tasso, C. (2011). A keyphrase-based paper recommender system. *Proceedings of Italian Research Conference (IRCDL 2011)*, 14-25.
http://dx.doi.org/10.1007/978-3-642-27302-5_2
- Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101, 5228-5235.
<http://dx.doi.org/10.1073/pnas.0307752101>
- Grün, B., & Hornik, K. (2011). Topicmodels: An R package for fitting topic models. *Journal of Statistical Software*, 40(13), 1-30. <http://dx.doi.org/10.18637/jss.v040.i13>
- He, Q., Pei, J., Kifer, D., Mitra, P., & Giles, L. (2010). Context-aware citation recommendation. *Proceedings of the 19th International Conference on World Wide Web*, 421-430.
<http://dx.doi.org/10.1145/1772690.1772734>
- Henning, V., & Reichelt, J. (2008). Mendeleey-a Last.fm for Research? *IEEE Fourth International Conference on eScience*, 327-328. <http://dx.doi.org/10.1109/escience.2008.128>
- Hornik, K., & Gruen, B. (2011). Topicmodels: an R package for fitting topic models. *Journal of Statistical Software*, 40(13), 1-30. *Topicmodels: an R package for fitting topic models*.
<http://dx.doi.org/10.1109/escience.2008.128>
- Huang, W., Kataria, S., Caragea, C., Mitra, P., Giles, C. L., & Rokach, L. (2012). Recommending citations: translating papers into references. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 1910-1914.
<http://dx.doi.org/10.1145/2396761.2398542>
- Hwang, S., & Chuang, S. (2004). Combining article content and Web usage for literature recommendation in digital libraries. *Online Information Review*, 28(4), 260-272.
<http://dx.doi.org/10.1108/14684520410553750>
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: with applications in R*. Springer.
- Jiang, Y., Jia, A., Feng, Y., & Zhao, D. (2012). Recommending academic papers via users' reading purposes. *Proceedings of the 6th ACM Conference on Recommender Systems*, 241-244.
<http://dx.doi.org/10.1145/2365952.2366004>
- Liang, T., Yang, Y., Chen, D., & Ku, Y. (2008). A semantic-expansion approach to personalized knowledge recommendation. *Decision Support Systems*, 45(3), 401-412.
<http://dx.doi.org/10.1016/j.dss.2007.05.004>
- Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: state

- of the art and trends. In F. Ricci, L. Rokach, B. Shapria & K. Paul (Eds.). Recommender systems handbook (pp. 73-105). Springer.
- Middleton, S. E., De Roure, D. C., & Shadbolt, N. R. (2001). Capturing knowledge of user preferences: ontologies in recommender systems. Proceedings of the 1st International Conference on Knowledge Capture (K-CAP 2001), 100-107.
<http://dx.doi.org/10.1145/500737.500755>
- Pudhiyaveetil, A., Kodakateri, G. S., Luong, H., & Eno, J. (2009). Conceptual recommender system for CiteSeerX. Proceedings of the third ACM Conference on Recommender Systems, 241-244. <http://dx.doi.org/10.1145/1639714.1639758>
- Sarkar, Deepayan (2008). Lattice: Multivariate data visualization with R. New York: Springer.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International Conference on World Wide Web, 285-295. <http://dx.doi.org/10.1145/371920.372071>
- Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapria & K. Paul (Eds.). Recommender Systems Handbook (pp. 257-297). Springer.
- Sugiyama, K., & Kan, M. (2010). Scholarly paper recommendation via user's recent research interests. Proceedings of Joint Conference of Digital Libraries. Gold Coast, 29-38.
<http://dx.doi.org/10.1145/1816123.1816129>
- Torres, R., McNee, S. M., Abel, M., Konstan, J. A., & Riedl, J. (2004). Enhancing digital libraries with TechLens+. Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries, 228-236. <http://dx.doi.org/10.1145/996350.996402>
- Vellino, A. (2010). A comparison between usage-based and citation-based methods for recommending scholarly research articles. Proceedings of the American Society for Information Science and Technology, 1-2. <http://dx.doi.org/10.1002/meet.14504701330>
- Wallach, H. M. (2008). Structured topic models for language. Ph.D. diss. University of Cambridge, UK.

• 국문 참고문헌에 대한 영문 표기
 (English translation of references written in Korean)

- Choi, Hoyeon, Shin, Dongwook, Choi, Joongmin, & Kim, Jungsun (2013). PARuS: Personalized academic paper recommender using semantic relation in digital library domain. Journal of

KISS: Software and Applications, 40(3), 164-175.

Park, Sang Jin, Kim, Yoon Hyun, & Lee, Ji-Hyun (2011). A recommender of academic papers using the citation analysis. The 35th Conferences of the KIPS, 18(1), 279-282.

Yeo, Woon-Dong, Park, Hyun-Woo, Kwon, Young-Il, & Park, Young-Wook (2010). Application of research paper recommender system to digital library. The Korean Contents Society, 10(11), 10-19. <http://dx.doi.org/10.5392/jkca.2010.10.11.010>