

A Folksonomy Ranking Framework: A Semantic Graph-based Approach*

Hyunjung Park**, Sangkyu Rho***

In collaborative tagging systems such as Delicious.com and Flickr.com, users assign keywords or tags to their uploaded resources, such as bookmarks and pictures, for their future use or sharing purposes. The collection of resources and tags generated by a user is called a personomy, and the collection of all personomies constitutes the folksonomy. The most significant need of the folksonomy users is to efficiently find useful resources or experts on specific topics. An excellent ranking algorithm would assign higher ranking to more useful resources or experts.

What resources are considered useful in a folksonomic system? Does a standard superior to frequency or freshness exist? The resource recommended by more users with more expertise should be worthy of attention. This ranking paradigm can be implemented through a graph-based ranking algorithm. Two well-known representatives of such a paradigm are PageRank by Google and HITS(Hypertext Induced Topic Selection) by Kleinberg. Both PageRank and HITS assign a higher evaluation score to pages linked to more higher-scored pages. HITS differs from PageRank in that it utilizes two kinds of scores: authority and hub scores. The ranking objects of these pages are limited to Web pages, whereas the ranking objects of a folksonomic system are somewhat heterogeneous(i.e., users, resources, and tags). Therefore, uniform application of the voting notion of PageRank and HITS based on the links to a folksonomy would be unreasonable. In a folksonomic system, each link corresponding to a property can have an opposite direction, depending on whether the property is an active or a passive voice.

The current research stems from the idea that a graph-based ranking algorithm could be applied to the folksonomic system using the concept of mutual interactions between entities, rather than the voting notion of PageRank or HITS. The concept of mutual interactions, proposed for ranking the Semantic Web resources, enables the calculation of importance scores of various resources unaffected by link directions. The weights of a property representing the mutual interaction between classes are assigned depending on the relative

* This study was supported by Seoul National University Research Grant(for Humanities and Social Sciences).

** Corresponding author, Researcher, Institute of Management Research, Seoul National University, Korea

*** Professor, Graduate School of Business, Seoul National University, Korea

significance of the property to the resource importance of each class. This class-oriented approach is based on the fact that, in the Semantic Web, there are many heterogeneous classes; thus, applying a different appraisal standard for each class is more reasonable. This is similar to the evaluation method of humans, where different items are assigned specific weights, which are then summed up to determine the weighted average. We can check for missing properties more easily with this approach than with other predicate-oriented approaches.

A user of a tagging system usually assigns more than one tags to the same resource, and there can be more than one tags with the same subjectivity and objectivity. In the case that many users assign similar tags to the same resource, grading the users differently depending on the assignment order becomes necessary. This idea comes from the studies in psychology wherein expertise involves the ability to select the most relevant information for achieving a goal. An expert should be someone who not only has a large collection of documents annotated with a particular tag, but also tends to add documents of high quality to his/her collections. Such documents are identified by the number, as well as the expertise, of users who have the same documents in their collections. In other words, there is a relationship of mutual reinforcement between the expertise of a user and the quality of a document. In addition, there is a need to rank entities related more closely to a certain entity. Considering the property of social media that ensures the popularity of a topic is temporary, recent data should have more weight than old data. We propose a comprehensive folksonomy ranking framework in which all these considerations are dealt with and that can be easily customized to each folksonomy site for ranking purposes.

To examine the validity of our ranking algorithm and show the mechanism of adjusting property, time, and expertise weights, we first use a dataset designed for analyzing the effect of each ranking factor independently. We then show the ranking results of a real folksonomy site, with the ranking factors combined. Because the ground truth of a given dataset is not known when it comes to ranking, we inject simulated data whose ranking results can be predicted into the real dataset and compare the ranking results of our algorithm with that of a previous HITS-based algorithm.

Our semantic ranking algorithm based on the concept of mutual interaction seems to be preferable to the HITS-based algorithm as a flexible folksonomy ranking framework. Some concrete points of difference are as follows. First, with the time concept applied to the property weights, our algorithm shows superior performance in lowering the scores of older data and raising the scores of newer data. Second, applying the time concept to the expertise weights, as well as to the property weights, our algorithm controls the conflicting influence of expertise weights and enhances overall consistency of time-valued ranking. The expertise weights of the previous study can act as an obstacle to the time-valued ranking because the number of followers increases as time goes on. Third, many new properties and classes can be included in our framework. The previous HITS-based algorithm, based on the voting notion, loses ground in the situation where the domain consists of more than two classes, or where other important properties, such as "sent through twitter" or "registered as a friend," are added to the domain. Fourth, there is a big difference in the calculation time and memory use between the two kinds of algorithms. While the matrix multiplication of two matrices has to be executed twice for the previous HITS-based algorithm, this is unnecessary with our algorithm.

In our ranking framework, various folksonomy ranking policies can be expressed with the ranking factors combined and our approach can work, even if the folksonomy site is not implemented with Semantic Web languages. Above all, the time weight proposed in this paper will be applicable to various domains, including social media, where time value is considered important.

Keywords : Information Technology, Ranking, Folksonomy, Collective Tagging, Semantic Web, FSRank

폭소노미 사이트를 위한 랭킹 프레임워크 설계: 시맨틱 그래프기반 접근

박 현 정, 노 상 규

I. 서 론

딜리셔스(Delicious.com)나 플리커(Flickr.com)와 같은 협력적 태깅 시스템(collaborative tagging system)은 웹 사용자들이 북마크, 문서, 사진, 동영상과 같은 웹 자원을 조직화하고 공유할 수 있는 수단을 제공한다[Noll and Meinel, 2008]. 협력적 태깅 시스템을 통해 한 사용자가 임의의 자원에 태그를 달 때 그가 생성한 모든 자원과 태그의 모음(collection)을 그의 퍼소노미(personomy)라고 하며, 이러한 퍼소노미들의 모음이 폭소노미를 이룬다[Hotho *et al.*, 2006]. 현재 대부분의 시스템에서는 다른 사용자의 퍼소노미를 살펴볼 수 있으며, 자원을 클릭하면 해당 자원을 업로드한 사용자와 그들이 할당한 태그들을, 태그를 클릭하면 해당 태그를 어떤 사용자들이 어떤 자원에 할당했는지 등 다각적인 조회가 가능하다. 이 외에도 다른 사용자가 업로드 한 자원을 복사하고, 이것에 자신의 태그를 달 수 있는 추가적인 기능도 제공한다. 이러한 태깅 시스템은 특별한 사용 기술이 없어도 이용할 수 있어, 많은 사용자에게 의해 매우 짧은 기간 안에 광대한 양의 정보가 축적되었고 지금도 엄청난 정보들이 저장되고 있다. 폭소노미 시스템을 이용하는 사용자들의 가장 큰

니즈 중의 하나는 특정 주제와 관련된 자원이나 특정 도메인에 대한 전문가를 효율적으로 찾는 것이라고 할 수 있다. 그런데 검색을 지원하는 랭킹은 그리 쉬운 문제가 아니다. 대부분의 폭소노미 사이트에서는 빈도나 최신성에 따라 정렬한 리스트를 제공하고 있지만, 이러한 방법들은 자원이나 사용자에 대한 실질적인 랭킹을 제공한다고 보기가 어렵다[Wetzker *et al.*, 2008].

두 개의 자원에 똑같이 10명의 사람들이 같은 태그를 할당했다면 과연 어떤 자원을 상위에 랭킹할 것인가? 대부분의 사람들은 10명의 사람들 중 해당 주제에 대한 뛰어난 전문가가 많을수록 이들에 의해 추천된 자원이 더욱 유용한 자원이라고 판단할 것이다. 그렇다면 폭소노미 시스템에서는 어떤 사람을 전문가라 할 수 있을까? Noll *et al.*[2009]의 폭소노미 랭킹에 관한 연구에서는 특정 주제에 대한 전문성 정도는 자원의 질과 상호 강화(mutual reinforcement)관계가 있다는 것과 전문가는 다른 사람이 발견하기 전에 유용한 자원을 식별해내는 사람이어야 한다는 두 가지 주요 요인을 가정하였다. 즉, 인기 높은 많은 자원에 태그를 단 사람일수록, 그리고 인기 높은 자원을 초기에 많이 발굴해낸 사람일수록 전문가라는 것이다.

여기에서 인기 높은 많은 자원에 태그를 단 사람일수록 더욱 높은 점수를 받는다는 가정은, 높은 점수를 가진, 더 많은 페이지들에 의해 추천될수록 해당 페이지의 점수가 올라가는 구글의 페이지랭크[Brin *et al.*, 1998; Page *et al.*, 1998; Haveliwala, 1999]와, 이웃하는 페이지들끼리 권위 점수와 허브점수를 서로 주고 받으며 계산되는 Kleinberg의 HITS(Hypertext Induced Topic Selection)[Kleinberg, 1999]로 대표되는 그래프기반 랭킹 알고리즘과 통하는 면이 있다. 이러한 그래프기반 랭킹 알고리즘은 구글에 의해 수십억 웹 페이지에 대한 대규모 데이터를 현실적으로 처리할 수 있는 가능성이 입증되면서 이미 다양한 분야에 대한 적용 연구가 이루어져 왔다.

페이지랭크나 HITS이 기본 패러다임은 참조하는 문서에서 참조되는 문서로 향하는 하이퍼링크를 추천의 의미로 간주하는 투표 개념[Brin *et al.*, 1998; Page *et al.*, 1998; Haveliwala, 1999]이다. 무수한 웹 문서들은 이러한 하이퍼링크만으로 연결되어 있으므로 하이퍼링크의 방향이 암시하는 정보를 활용해 최고로 인기가 많은 문서를 찾는 것이 가능했다. 이와 대조적으로 폭소노미 시스템에는 사용자, 자원, 태그라는 세 가지 이질적인 개체들이 존재하며, 이들 간의 연결관계를 나타내는 링크들의 속성도 동일하지 않다. 특히, 각 링크속성을 수동태로 또는 능동태로 표현함에 따라 이에 대한 링크 방향은 정반대가 될 수 있다. 결국 링크 방향을 일률적인 투표 개념으로 인정하기 힘든 임의성이 크기 때문에 페이지랭크나 HITS를 폭소노미와 같은 이질적인 도메인에는 적용하기 어려운 한계점이 있다.

본 연구는 기존의 페이지랭크나 HITS의 투표 개념이 아닌, 개체간의 상호작용성 개념을 바탕으로 하는 그래프기반 랭킹 알고리즘을 폭소노미 시스템에 적용해보려는 아이디어로 출발하였다. 상호작용성의 개념은 시맨틱 웹의 다양한 자원들을 랭킹하기 위해 제안된[Park *et al.*, 2011] 개념으로 클래스와 클래스간 상호작용을 나타내

는 속성들의 가중치를 특정클래스에 속하는 개체의 중요도에 영향을 미치는 정도에 따라 결정하여, 링크의 방향과는 무관하게 다양한 개체들의 중요도를 구할 수 있게 해준다.

그런데, 폭소노미 시스템에는 Park *et al.*[2011]의 연구에서는 고려하지 않은 다음과 같은 특성들이 존재한다. 첫째, 한 주제에 대한 관심이 오래 지속되지 못하는 특성이 있어[Hayes *et al.*, 2007; Hayes and Avesani, 2007] 오래된 데이터보다는 최신 데이터에 더 높은 점수를 부여해야 한다. 둘째, 같은 자원에 비슷한 태그를 설정한 사용자들이 많을 경우 태그를 설정한 순서에 따라 사용자의 전문성 점수를 차등 부여할 필요가 있다[Noll *et al.*, 2009]. 셋째, 폭소노미에서는 대부분 한 사용자가 같은 자원에 여러 개의 태그들을 설정하거나, 서로 다른 사용자들이 같은 자원에 같은 태그들을 설정할 수 있으므로 같은 개체 쌍에 대한 관계가 중복적으로 발생한다[Klyne *et al.*, 2004]. 넷째, 특정 개체와 관계가 있는 개체들을 관계의 강도에 따라 랭킹해 봄으로써 특정 개체의 커뮤니티를 분석해볼 필요가 있다[Hotho *et al.*, 2006].

본 논문은 이러한 이슈들에 대한 해결 방안으로서 Park *et al.*[2011]의 연구를 더욱 발전시켜 폭소노미 개별 사이트의 랭킹 용도에 맞게, 속성 가중치를 커스터마이징하여 사용할 수 있는 종합적인 랭킹 프레임워크를 제안한다. 상호작용성을 기반으로 폭소노미 랭킹 니즈를 랭킹 팩터로 반영하기 위한 방안의 타당성을 검증하기 위해 모의실험을 수행하였고, 무수한 기존 연구들이 추구해 온 투표 개념 기반 알고리즘과의 비교 분석을 위해 폭소노미 전문가 랭킹 알고리즘인 SPEAR[Noll *et al.*, 2009]와 본 논문에서 제안하는 알고리즘으로 딜리셔스 실제 데이터를 랭킹하였다. 본 논문에서 제안하는 방법은 상호작용성에 새롭게 시간 가치를 적용함으로써 SPEAR의 핵심이라고 할 수 있는 발견자와 추종자 가중치를 더욱 탄력적으로 조정할 수 있고, 기본적인 속성 가중치에

도 시간 가치를 적용함으로써 최신 인기 데이터를 보다 효과적으로 상위에 랭킹 할 수 있다. 또, 커뮤니티 분석 용도로도 활용될 수 있다. 본 논문에서 최초로 제안되는 시간 가치의 적용 방법은 소셜 미디어와 같이 최신 데이터를 중요시하는 다른 도메인에도 적용할 수 있을 것으로 보인다.

본 논문의 구성은 다음과 같다. 먼저, 제 II장에서 이론적인 배경 및 관련 연구들에 대해 살펴보고, 제 III장에서는 폭소노미 랭킹에서 고려해야 할 주요요인들에 대해 더욱 구체적으로 논의한다. 그리고 제 IV장에서는 이러한 주요요인들을 랭킹 팩터로 반영하였을 때 결과의 타당성 확인을 위한 모의실험을 한 후, 실제 폭소노미 사이트에 대한 랭킹 결과를 분석한다. 마지막으로, 제 V장에서 연구내용을 정리하고 연구의 한계점 및 미래 연구방향에 대해 논의한다.

II. 이론적 배경 및 관련 연구

2.1 폭소노미(Folksonomies)

협력적 태깅 시스템[Hammond *et al.*, 2005]은 임의의 사용자들이 웹 상에서 이용 가능한 수많은 자원들에 대해 자유롭게 태그를 할당하도록 해준다. 서로 다른 사용자들이 의해 수집되고 다양한 태그가 붙여져 상향식(bottom-up)으로 분류된 자원들이 모이면 폭소노미(folksonomies)가 탄생하는데, 하나의 폭소노미(a folksonomy)는 사용자, 자원, 태그 및 이들 간의 태그할당 관계로서 다음과 같이 정의된다.

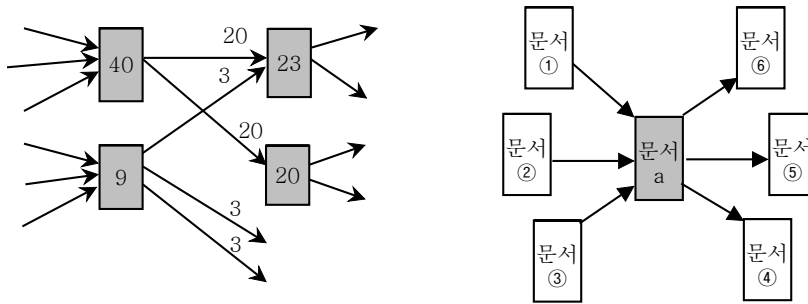
정의: 하나의 폭소노미(a folksonomy)는 사용자 U , 자원 R , 태그 T 와 이들 간의 태그할당(tag assignment)을 나타내는 삼항관계(ternary relation)인 $Y(Y \subseteq U \times R \times T)$ 로 이루어진 하나의 튜플(tuple) $F=(U, R, T; Y)$ 이다.

여기에서 사용자는 웹 상의 자원에 태그를 할당하는 사람이고, 태그는 임의의 문자열이며, 자원은 딜리셔스에서는 URL, 플리커에서는 그림(picture), 유튜브(YouTube)에서는 동영상 등으로 시스템의 유형에 따라 달라진다. 사용자, 자원, 태그 간의 삼항관계는 삼자 하이퍼그래프(tripartite hypergraph)로 표현된다[Mika, 2007](<그림 3> 좌측 참조).

2.2 관련 연구

2.2.1 그래프기반 랭킹 알고리즘들

문서와 문서가 하이퍼링크로 연결된 월드와이드웹(World Wide Web)에서는 페이지랭크나 HITS와 같이, 거대한 웹 그래프의 문서 간 링크 구조를 분석하여 문서의 중요도 점수를 산출하는 방법이 사용되어 왔다. 1998년에 등장하여 주목을 받아 온 구글의 페이지랭크[Brin *et al.*, 1998; Page *et al.*, 1998; Haveliwalla, 1999]에서는 임의의 페이지를 가리키는 다른 페이지들이 많을수록, 그리고 이러한 다른 페이지들의 중요도가 높을수록 해당 페이지의 중요도가 올라간다. <그림 1>의 좌측 단순 페이지랭크에서는 참조하는 웹 문서의 중요도가 각 링크에 균등 배분되며, 참조되는 문서의 중요도는 자신을 가리키는 링크에 실린 중요도들의 합으로 결정된다. Kleinberg의 HITS[Kleinberg, 1999]에서는 하나의 웹 페이지에 권위(authority)와 허브(hub)라는 두 가지 종류의 점수가 계산되며, 다른 중요한 페이지들에 의해 링크되는 정도가 강할수록 권위점수가 높아지고, 다른 중요한 페이지들을 링크하는 정도가 강할수록 허브점수가 올라간다. <그림 1>의 우측 HITS에서 문서 a의 권위점수는 문서 a를 가리키는 문서 1, 2, 3의 허브점수의 합으로, 문서 a의 허브점수는 문서 a가 가리키는 문서 6, 7, 8의 권위점수의 합으로 계산된다.



<그림 1> 단순페이지랭크(좌)와 HITS(우)

이 후 이메일 이용자들을 노드로, 이들간의 이메일 교환관계를 링크로 하는 그래프를 대상으로 특정 주제에 대한 전문성에 따라 이메일 이용자들을 랭킹하는 것에 관한 연구[Dom *et al.*, 2003], 자바 포럼(Java Forum)과 같이 전문성을 공유하려는 커뮤니티에서 커뮤니티 이용자들을 나타내는 노드와 질문을 올린 사람으로부터 이에 대한 답을 올리는 사람들로 향하는 링크로 이루어진 그래프에 페이지랭크를 응용한 그래프기반 방법을 적용하여 전문성을 랭킹하는 것에 관한 연구[Zhang *et al.*, 2007]가 있었다. 그리고 사람과 다른 자원을 혼합하여 하나의 네트워크 그래프를 구성하고 이에 대해 그래프기반 랭킹 알고리즘을 적용한 연구도 있었는데, Zhou *et al.*[2007]은 저자와 논문을 노드로, 논문 저자간의 공동저술관계, 논문간의 참조관계, 저자와 논문간의 저술관계를 링크로 하는 이질적인(heterogeneous) 그래프에 대해 랭킹 알고리즘을 적용하였고, Wang *et al.*[2002]은 웹 페이지와 웹 이용자를 노드로, 웹 이용자가 웹 페이지를 방문하는 관계와 웹 페이지 간의 참조관계를 링크로 하는 그래프에서 특정 주제에 대한 관련도(relevance)를 랭킹하는 것에 대해 논의하였다.

Abel *et al.*[2008]은 그룹미(GroupMe) 폭소노미 시스템에서 사용자들이 드래그앤드랍(drag and drop)으로 관련 있는 웹 자원들을 그룹으로 조직화할 수 있음을 고려하여, 포크랭크(FolkRank)[Hotho *et al.*, 2006], 소셜심랭크(SocialSimRank)

[Bao *et al.*, 2007], 소셜페이지랭크(SocialPageRank)[Bao *et al.*, 2007]와 이들을 보완한 알고리즘들에 대한 분석결과를 제시하였다. 또, Orlicki *et al.*[2009]은 유튜브, 플리커와 같은 협력적 태깅 시스템에서 사용자를 노드, 멀티미디어 추천관계를 태그가 달린 링크로 하는 태그 그래프에 대해, 패싯(facet)이라 불리는 태그 셋(set)으로 이루어진 검색 질의에 대한 사용자 랭킹의 온라인 계산문제를 다루었다. 이들은 각 태그와 관련된 서브그래프에 대해 먼저 오프라인으로 페이지랭크 방식의 랭킹을 계산해놓고, 패싯에 있는 모든 태그에 대한 랭킹 결과를 온라인 상에서 종합하여 최종 랭킹을 산출하는 알고리즘들과 이들에 대한 분석결과를 제시하였다. 이 외에도 폭소노미를 대상으로 그래프기반 랭킹 알고리즘을 적용한 연구에는 Hotho *et al.*[2006]의 포크랭크(FolkRank)와 Noll *et al.*[2009]의 스피어(SPEAR)가 있는데 이 연구들은 본 연구와 관련도가 높으므로 다음 절에서 좀 더 자세히 설명하겠다.

2.2.2 폭소노미 커뮤니티 분석을 위한 포크랭크(FolkRank)

Hotho *et al.*[2006]은 중요한 많은 페이지들에 의해 참조될수록 해당 페이지의 중요도가 올라가는 페이지랭크[Brin and Page, 1998]의 기본 개념을 폭소노미에 적용하여, 어댑티드(adapted) 페이지랭크를 고안하고, 이를 이용하여 포크랭크라

는 새로운 알고리즘을 제안하였다. 어댑티드 페이지랭크에서는 중요한 사용자에게 의해 중요한 태그가 할당될수록 해당 자원의 중요도가 올라가며, 이러한 성질은 태그나 사용자에게 대해서도 대칭적으로 유사하게 적용된다. 어댑티드 페이지랭크를 구하기 위해서는 먼저, 방향성 없는(undirected) 삼자 하이퍼그래프(tripartite hypergraph)로 표현되는 폭소노미 그래프를 사용자, 자원, 태그를 노드로 하고 이들 간의 연결관계를 링크로 하는 그래프로 바꾸고, 각 링크의 가중치는 해당 링크 연결관계의 빈도(frequency)에 비례하는 값으로 설정한다. 예를 들어, 사용자 u 와 태그 t 를 연결하는 링크의 가중치는 u 가 t 를 할당한 자원들의 수에 비례하고, 사용자 u 와 자원 r 을 연결하는 링크의 가중치는 u 가 r 에 할당한 태그의 수에 비례한다. 포크랭크는 특정 개인이 선호하는 페이지에 높은 가중치를 부여하여 개인 중심의 페이지랭크를 구하는 개인화된(personalized) 페이지랭크[Page *et al.*, 1998]를 사용자나 자원 또는 태그 개체에 적용하여 구한 개인화된 어댑티드 페이지랭크 점수와 개인 차이를 무시한 어댑티드 페이지랭크 점수의 차이로 계산된다. 포크랭크의 장점은 사용자, 자원, 태그 중 특정 개체에 높은 가중치를 부여한 후, 다른 개체들의 점수 증감을 분석함으로써 해당 개체와 연결관계가 높은 개체들을 선별할 수 있다는 것이다.

2.2.3 폭소노미 전문가 랭킹을 위한 스피어 (SPEAR)

Noll *et al.*[2009]은 특정 주제에 대한 유저의 전문성 정도는 자원의 질(quality)과 상호 강화(mutual reinforcement) 관계가 있다는 것과 전문가는 다른 사람들이 발견하기 전에 유용한 자원을 식별해내는 사람이어야 한다는 두 가지 주요 요인을 가정하고, 이러한 아이디어를 구현한 그래프기반 유저 랭킹 알고리즘인 SPEAR(Spamming-resistant Expertise Analysis and Ranking)를 제

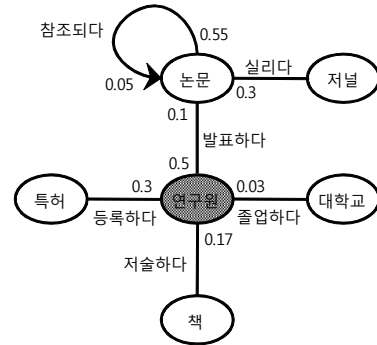
안하였다. SPEAR는 HITS 알고리즘[Kleinberg, 1999]과 비슷하게 인접 개체의 점수를 서로 더하는 방식으로 계산되는데, 가중치행렬을 만들 때 사용자관련 가중치가 단순한 연결관계를 표시하는 0 또는 1이 아니라 발견자(discoverer)와 추종자(follower)의 개념을 적용하여 구한 함수로 설정된다. HITS 방식으로는 유용한 자원에 대해 최초 발견자나 초기에 태그를 단 사용자와 이 자원이 이미 인기가 높아진 후 태그만을 단 사용자의 전문성을 똑같이 평가하지만, SPEAR는 자원의 초기 설정자일수록 추종자보다 더 높은 점수를 부여한다. 이것은 특정 사용자와 자원 간의 링크에 대한 가중치를 특정 사용자가 태그를 단 후, 같은 자원에 같은 태그를 단 다른 사용자들 수의 루트함수로 설정함으로써 구현된다. SPEAR에서 사용자들은 허브점수(hub score), 자원은 권위점수(authority score)를 가지며, 특정 태그에 연결된 사용자와 자원들을 미리 추출하여 생성된 폭소노미 서브셋(subset)에 대해 계산되므로 태그의 랭킹 점수는 구하지 않는다. 이들은 SPEAR 알고리즘이 HITS나 빈도(frequency) 기준방법 보다 다양한 유형의 전문가들을 찾아내는데 있어 우수하며, 초기 발견자의 점수는 올리고 인기를 얻은 자원에 태그를 달으로써 쉽게 점수를 획득하려는 스팸머들의 점수는 상대적으로 떨어뜨리는 효과가 있음을 보여주었다.

2.2.4 상호작용성에 기반하여 다양한 시맨틱 웹 자원을 랭킹하기 위한 연구

Park *et al.*[2011]은 다양한 시맨틱 웹 자원(resource)의 중요도를 RDF(Resource Description Framework) 링크 방향과 무관한 자원간의 상호작용성을 바탕으로 계산하는 방안을 제안하고 그 수렴성을 증명하였다. 상호작용성은 이웃하는 다른 개체가 해당 개체의 중요도를 계산하는데 미치는 영향력과 해당 개체가 다른 개체의 중요도에 미치는 영향력의 상호관계를 일컫는 말이다.

속성의 가중치 크기를 결정할 때에는 특정 개체의 중요도를 계산하는데 다른 개체들이 얼마만큼 중요하게 영향을 미치는지를 분석하여 결정한다. 그들은 이렇게 자원의 중요도에 영향을 미치는 속성의 가중치를 특정 클래스에 대한 상대적인 비중을 고려하여 결정함으로써 기존 그래프 기반 알고리즘의 단점인 강한 결합 모임 현상 (Tightly Knit Community Effect)을 효과적으로 해결할 수 있음을 보였다. 이러한 상호작용성에 의한 중요도 계산 방식은 사람들의 평가체계와도 상통하는 면이 있다. <그림 2>에서 연구원의 중요도는 어떤 논문들을 발표했느냐를 50%, 어떤 특허를 등록했느냐를 30%, 어떤 책을 저술했느냐를 17%, 어떤 대학교를 졸업했느냐를 3% 반영하여 결정된다. 또, 논문의 중요도는 어떤 논문들에 의해 참조되었는지를 55%, 어떤 저널에 실렸는지를 30%, 어떤 연구원이 발표했는지를 10%, 어떤 논문들을 참조했는지를 5% 반영하여 결정된다. 이것은 전체 그림의 일부분이라 모든 클래스와 속성들이 표현되어 있지 않지만, 특허, 책, 대학교 등도 연구원 클래스처럼 각각의 중요도에 영향을 미치는 속성에 대한 가중치를 가지고 있다. 속성의 종류와 가중치의 크기는 해당 도메인을 위한 어플리케이션의 용도에 따라 달라질 수 있다. 이렇게 Park *et al.*[2011]의 연구에서는 도메인을 구

성하는 클래스와 클래스 또는 자원과 자원간의 상호작용을 유의미한 속성들로 기술하고, 서로에게 미치는 영향력에 따라 가중치를 부여한다.

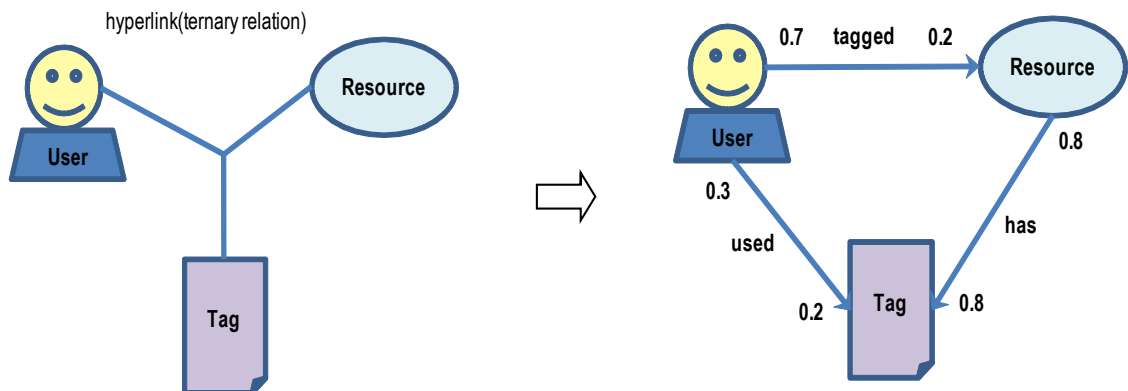


<그림 2> 상호작용성 기반 시맨틱 랭킹 계산 예

Ⅲ. 폭소노미 랭킹의 핵심 개념과 알고리즘

3.1 도메인 구성 및 속성 가중치

<그림 3>의 좌측에 있는 폭소노미 하이퍼그래프 <그림 3>의 우측에 있는 노드 $V = U \cup R \cup T$ 인 RDF[Klyne *et al.*, 2004; Manola *et al.*, 2004] 그래프로 변환이 가능하다. 삼항관계(ternary relation)인 Y 에 의해 사용자와



<그림 3> 폭소노미 하이퍼그래프(좌)의 RDF 그래프(우) 변환 및 속성가중치 설정 예

자원, 사용자와 태그, 자원과 태그 트리플이 생성될 때마다 앞의 개체를 주어부, 뒤의 개체를 목적부 노드로 하는 링크가 만들어지며, 링크 집합 E 는 다음과 같이 표현될 수 있다. 여기에서 (u, r) 은 u 를 주어부, r 을 목적부로 하는 링크를 나타낸다.

$$E = \{(u, r) | \exists t \in T: (u, r, t) \in Y\} \\ \cup \{(u, t) | \exists r \in R: (u, r, t) \in Y\} \\ \cup \{(r, t) | \exists u \in U: (u, r, t) \in Y\}$$

결과적으로, 폭소노미 도메인은 세 가지 클래스와 세 가지 속성으로 간단하게 표현될 수 있는데, 실제 인스턴스 데이터는 <그림 3>의 RDF 그래프 같은 삼각형 모양의 관계가 복잡하게 얽혀 있다. 링크 옆에 표시되어 있는 가중치는 Park *et al.*[2011]의 시맨틱 웹 자원랭킹 알고리즘을 폭소노미 도메인에 적용하여 임의로 설정한 것이며, <표 1>은 <그림 3>의 RDF 그래프를 (주어부, 속성, 목적부)로 이루어진 RDF 트리플 및 주어부 가중치(subWt), 목적부 가중치(objWt) 순으로 정리한 것이다. 주어부 가중치는 주어부 자원의 중요도를 구할 때, 목적부 자원의 중요도에 곱해지는 가중치를 의미하며, 목적부 가중치도 이와 같은 방식으로 사용된다. 상호작용성을 바탕으로 하는 Park *et al.*[2011]의 연구에서는 링크 방향과 독립적으로 자원의 중요도를 계산하지만, 하나의 링크에 두 가지 가중치가 있으므로, 설명의 편의상 링크 방향을 이용하여 주어부 가중치와 목적부 가중치로 구분한다.

<표 1> 폭소노미 주어부/목적부 속성가중치

	RDF 트리플			속성 가중치	
	주어부 (subject)	속성 (predicate)	목적부 (object)	주어부가중치 (subWt)	목적부가중치 (objWt)
1	expert(user)	tagged	resource	0.7	0.2
2	expert(user)	used	tag	0.3	0.2
3	Resource	has	tag	0.8	0.8

3.2 자원의 발견자(discoverer)와 추종자(follower)

협력적 태깅 시스템에서 전문성을 평가하는 가장 단순한 방법은, 현재 많은 시스템에서 사용하고 있는 것처럼, 해당 주제에 속하는 태그의 사용 빈도에 의한 방법일 것이다. 그런데, 양(quantity)이 질(quality)을 대변하지 못하고, 무분별하게 태그를 다는 스팸머들(spammers)도 있어 이상적인 방법은 아니라고 할 수 있다. 전문성은 목적을 달성하기 위해 가장 필요한 정보를 선택하는 능력을 포함한다는 사실을 발견한 심리학 연구가 있다[Feltovich *et al.*, 2006]. 앞에서 언급한 것처럼, HITS와 같은 접근방법으로는 처음으로 자원을 발견하고 태깅을 한 사용자와 이미 인기가 높아진 자원에 태깅만 한 사용자의 전문성을 같은 수준으로 평가한다. 그렇다고 Noll *et al.*[2009]의 방식을 그대로 사용하면, 오래 전에 태깅 한 사용자일수록 추종자가 누적되어 더 높은 가중치를 받게 될 것이므로, 다음 절에서 설명될 시간 가중치와 기본적으로 상충되는 면이 있다. 그래서 본 논문에서는 <표 1>의 (expert/user, tagged, resource)라는 트리플의 주어부 가중치를 설정할 때, 추종자 수가 아닌, 액션 시점을 반영한 'tagged' 속성의 주어부 가중치들을 더한 값을 사용하여 오래된 추종자들의 영향력을 감소시켰다.

3.3 시간 가중치

블로그스피어에는 한 주제에 대한 블로거들의

관심이 오래 지속되지 못하는 동적인 특성이 있다[Hayes *et al.*, 2007; Hayes and Avesani, 2007]. 이러한 특성은 다른 소셜 미디어 서비스들에 대해서도 비슷하게 나타나고 있다. 그런데, Hotho *et al.*[2006]과 Noll *et al.*[2009]을 포함하는 기존의 모든 그래프기반 랭킹 알고리즘에서는 오래 전에 발생한 이벤트와 최근에 발생한 것을 똑같이 취급하므로 오래 전에는 적극적으로 활동했지만 최근에는 별로 활동하지 않는 사용자나 그 반대의 경우 등을 가려내지 못한다.

본 연구에서는 이러한 문제점을 고려하여 이벤트가 발생한 시각에 따라 가중치를 달리하는 방안을 제시한다. 다시 말하면, <표 1>의 속성 가중치는 현재 시점을 기준으로 하는 값들이고, 오래 전에 발생한 이벤트일수록 이에 대한 가중치는 현재 시점 가중치 보다 더 많이 감소되도록 하자는 것이다. 이러한 시간 가중치 함수는 도메인 특성에 따라 다양한 곡선 형태로 표현될 수 있겠지만, 본 논문에서는 결과에 대한 분석이 용이하도록 타임구간에 따라 계단형으로 감소하는 단순한 형태를 가정하였다. <표 2>는 ‘타임구간 2’에 해당하는 데이터들에는 <표 1>의 가중치의 2/3를, ‘타임구간 3’에 해당하는 데이터들에는 1/3을 적용하도록 하는 것을 나타낸다. 하나의 타임구간은 컨텍스트에 따라 일주일이나 한 달 등이 될 수 있다.

3.4 FSRank(Folksonomy Semantic Rank) 알고리즘

본 논문에서는 사용자가 자원에 태깅을 한 액션 시점에 대한 타임 정보를 추가하여, 이것을 삼항관계(ternary relation)인 $Y(Y \subseteq U \times R \times T)$ 에 시간을 나타내는 첨자 t 를 붙여 $F = (U, R, T; Y_t)$ 로 표기하기로 한다. 비슷한 방식으로 폭소노미 $F = (U, R, T; Y_t)$ 를 RDF 그래프로 변환할 때에도 링크 E 에 타임 첨자 t 를 붙여, 노드(V)와 타임 링크(E_t)로 이루어진 폭소노미 RDF 그래프 $G_F = (V, E_t)$ 로 표기하도록 한다.

본 논문에서 제안하는 폭소노미 랭킹 알고리즘인 FSRank(Folksonomy Semantic Rank)의 전체적인 흐름은 <그림 4>와 같고, 알고리즘 과정 중 가중치 w_{rk} 의 설정 방법은 <그림 5>와 같다. w_{rk} 는 노드 r 의 랭킹 점수를 계산할 때 노드 k 의 점수에 곱해지는 가중치이며, 전체 노드의 수는 $N = n(V)$ 이다. g^r 은 노드 $r(1 \leq r \leq N)$ 의 랭킹 점수를 나타내고, $g^{\cdot r}$ 은 노드 $r(1 \leq r \leq N)$ 의 정규화되기 이전 랭킹 점수이다. 위 첨자가 붙지 않고, 아래 첨자만 붙은 g_i 는 전체 N 개 노드의 랭킹 점수를 나타내는 $(N \times 1)$ 벡터이며, 아래 첨자 i 는 이터레이션 회수를 나타낸다. 실제 연산은 초기화 벡터인 1벡터와 가중치 w_{rk} 를 원소로 갖는 가중치 행렬을 곱하고, 결과로 얻어진 벡터를 다시 가중치 행렬과 곱하는 이터레이션을 m 번 반

<표 2> 폭소노미 속성에 대한 시간가중치 적용 예

	RDF 트리플			속성 가중치 및 타임구간별 적용 방법			
	주어부 (subject)	속성 (predicate)	목적부 (object)	주어부가중치 (subwt)	목적부가중치 (objwt)	타임구간 2	타임구간 3
1	expert(user)	tagged	resource	0.7	0.2	$\times 2/3$	$\times 1/3$
2	expert(user)	used	tag	0.3	0.2	$\times 2/3$	$\times 1/3$
3	resource	has	tag	0.8	0.8	$\times 2/3$	$\times 1/3$

복하는 형태로 이루어진다. 이러한 형태의 알고리즘에 대한 수렴성 분석은 Park *et al.*[2011]에서 이미 수행된 바 있다.

- ① 초기화(Initialization): $g_0^r = 1 (1 \leq r \leq N)$.
- ② 이터레이션(Iteration): $i = 1, 2, \dots, m$ 에 대해 다음을 반복한다.
 - a. 각 노드 r 에 대해, 다음 식을 계산한다.

$$g_i^r = \sum_k g_{i-1}^k \times w_{rk}$$

- b. g_i^r 를 정규화하여 g_i 를 얻는다. 정규화 조건은 다음 식과 같다.

$$\sum_r (g_i^r)^2 = 1.$$

- ③ 반환(Return): g_m 을 반환한다.

<그림 4> FSRank 알고리즘의 전체적인 흐름

<그림 5>에서 추종자 가중치 f_k 는 트리플 k 의 주어부가 사용자, 목적부가 자원일 때 트리플 k 의 액션 타임 후에 트리플 k 의 자원과 같은 자원에, 트리플 k 의 사용자와 같은 태그를 한 개 이상 설정한 사용자에게 대해, 해당 트리플의 액션 타임(t)에 해당하는 속성 가중치를 누적 합산하여 계산된다. 여기에서 한 사용자가 임의의 자원에 동시에 2개 이상의 태그를 설정한 경우에 대한 트리플들은 모두 하나로 간주된다. f_k 는 유용한 자원을 먼저 발견한 사용자들의 점수는 높이고, 이미 인기가 높아진 자원에 태그만을 설정하려는 스팸머들의 점수는 상대적으로 낮추기 위한 것이다. 그런데, 추종자의 수를 단순히 더하여 계산하면, 시간이 경과될수록 추종자의 수가 누적되어, 오래 전의 발견자에게 너무 큰 가중치가 부여되는 문제점이 있어 시간을 고려한 속성 가중치로 합산하는 것이다. 추종자 점수는 폭소노미 사이트의 랭킹 정책에 따라, 최소한 기준 수 이상의 추종자들이 있을 때에 정해진 등수까지의 사용자들에게만 전문가 점수를 부여하는 등으로 조정하여 반영할 수도 있다.

- ① 폭소노미의 세 가지 트리플 속성에 대한 속성 가중치와 타임구간에 대한 가중치 함수를 설정한다.
- ② 폭소노미 $F = (U, T, R; Y_t)$ 에서 특정 주제에 대한 태그 셋(T_s)에 해당하는 서브 그래프 $F = (U, T_s, R; Y_t)$ 를 추출한다.
- ③ $F = (U, T_s, R; Y_t)$ 에 대한 폭소노미 RDF 그래프 $G_F = (V, E_t)$ 를 타임 스탬프가 추가된 트리플 형식으로 변환한다.
- ④ 트리플 셋에 대해 순차적으로 해당 가중치 행렬 셀에 들어갈 가중치를 계산하여 기록하되, 주어부와 목적부가 같아 동일한 셀에 중복되는 가중치들은 누적 합산한다. 이때, 주어부가 사용자이고 목적부가 자원인 경우에는 식 (1)과 같이, 그 외의 경우에는 식 (2)와 같이 계산한다.

$$w_{ij} = \left(\sum_{k \in P_{ij}} subTWt_k \right) \times f_k, \quad w_{ji} = \sum_{k \in P_{ij}} objTWt_k \quad (1)$$

$$w_{ij} = \sum_{k \in P_{ij}} subTWt_k, \quad w_{ji} = \sum_{k \in P_{ij}} objTWt_k \quad (2)$$

- P_{ij} : 주어부가 노드 i 이고, 목적부가 노드 j 인 타임 스탬프 트리플의 중복집합(multiset, bag)
- $subTWt_k$: 트리플 k 의 액션 타임이 반영된 주어부 가중치
- $objTWt_k$: 트리플 k 의 액션 타임이 반영된 목적부 가중치
- f_k : 트리플 k 에 대한 추종자를 각각의 타임 반영 속성 가중치로 합산한 값 반영, 추종자가 없을 때 0이 되지 않도록 1을 더해줌

<그림 5> 가중치 행렬의 각 셀에 들어가는 가중치 계산 방법

<그림 5>의 계산 과정에 대한 구체적인 예로 <표 3>을 보면, 5회의 폭소노미 관련 액션이 시간 순으로 정렬되어 있다. 오늘이 2011년 5월 7일이고 오늘을 기준으로 2개월 전 데이터들에는 2/3배의 가중치가 적용된다고 가정하자. 1행은 2011년 3월 1일에 사용자 u_1 이 URL이나 사진 등의 자원인 r_1 에 태그 t_1 과 t_2 를 설정한 것을 의미한다. 나머지 행들도 모두 자원 r_1 에 태그 t_1 과 t_2 를 설정했으므로, 특정 사용자 보다 뒤에 오는 사용자들은 모두 해당 특정 사용자의 추종자들이라고 할

수 있다. <그림 5> 식 (1)의 추종자 가중치 f_k 를 구하기 위해서는 먼저 (user u , tagged, resource r)에 대한 타임 반영 주어부 가중치를 알아야 한다. <표 2>에서 기본적인 주어부 가중치는 0.7이고, 음영 처리된 1~3번 행은 모두 2개월 전 데이터이므로 이들에 대한 타임반영 속성가중치를 계산할 때에는 모두 원래의 가중치에 $2/3$ 가 곱해진다. 다음으로 1번 행에 대한 추종자 가중치는 1번 보다 뒤에 오는 2~5번의 타임반영 주어부 속성가중치들을 더한 값에 숫자 1을 더해서 계산된다. 트리플 변환과정은 생략하였지만, 1번 행을 트리플로 변환하면 (u1, used, t1), (u1, used, t2), (r1, has, t1), (r1, has, t2), (u1, tagged, r1), (u1, tagged r1)이라는 6개의 트리플이 생성된다. 사용자 u1을 주어부로, 자원 r1을 목적부로 하는 트리플이 2개 있으므로 1번 행에 의해 가중치 행렬 (u1, r1) 셀에 기록되는 최종 사용자 가중치는 타임반영 속성가중치에 2배를 하고 이것에 추종자 가중치를 곱한 3.11이 된다. 이와 대칭 위치인(r1, u1) 셀에 기록되는 가중치는(u1, tagged, r1)의 타임반영 목적부 가중치를 두 배 한 $2 \times (0.2 \times 2/3) \approx 0.27$ 이 된다. 1번 행에 의해(r1, t1) 셀에 기록되는 가중치는(r1, has, t1)의 주어부 가중치가 0.8이므로 $0.8 \times 2/3$ 로 계산되며, 1~5번 전체 폭소노미에 대해 최종적으로 기록되는 값은 1~3번에 대한

$3 \times (0.8 \times 2/3)$ 와 4~5번에 대한 2×0.8 을 합한 3.2가 된다. <그림 6>은 전체 폭소노미 데이터가 <표 3>이라고 가정할 때의 최종 가중치행렬이다. HITS방식대로 계산되는 SPEAR의 경우에는 속성가중치나 시간가중치의 개념이 없고, 사용자에서 자원으로 향하는 링크에 대한 추종자 가중치는 추종자 수의 루트함수이므로, 행 순서대로 $\sqrt{5}$, $\sqrt{4}$, $\sqrt{3}$, $\sqrt{2}$, 1이고, 자원에서 사용자로 향하는 링크의 가중치는 연결이 있으면 무조건 1, 없으면 0으로 기록된다.

	u1	u2	u3	u4	u5	r1	t1	t2
u1	0	0	0	0	0	3.11	0.2	0.2
u2	0	0	0	0	0	2.68	0.2	0.2
u3	0	0	0	0	0	2.24	0.2	0.2
u4	0	0	0	0	0	2.38	0.3	0.3
u5	0	0	0	0	0	1.4	0.3	0.3
r1	0.27	0.27	0.27	0.4	0.4	0	3.2	3.2
t1	0.13	0.13	0.13	0.2	0.2	3.2	0	0
t2	0.13	0.13	0.13	0.2	0.2	3.2	0	0

<그림 6> FSRank의 <표 3>에 대한 가중치 행렬

IV. 실험 및 평가

본 논문에서 제안하는 FSRank의 타당성 및 성능

<표 3> (user u , tagged, resource r)트리플의 주어부인 사용자 가중치 계산과정 예

	Time Stamp	u	r	t	타임반영 속성 가중치(subTW)	추종자 가중치(f)	최종 사용자 가중치
1	2011-03-01	u1	r1	t1, t2	$0.7 \times 2/3$ = 7/15	$1 + 0.7 \times 2/3 + 0.7 \times 2/3 + 0.7 + 0.7$ = 10/3	$(2 \times 7/15) \times 10/3$ = $28/9$ = 3.11
2	2011-03-02	u2	r1	t1, t2	$0.7 \times 2/3$ = 7/15	$1 + 0.7 \times 2/3 + 0.7 + 0.7$ = 43/15	$(2 \times 7/15) \times 43/15$ = $602/225$ = 2.68
3	2011-03-03	u3	r1	t1, t2	$0.7 \times 2/3$ = 7/15	$1 + 0.7 + 0.7$ = 2.4	$(2 \times 7/15) \times 2.4$ = $56/25$ = 2.24
4	2011-04-15	u4	r1	t1, t2	0.7	$1 + 0.7$ = 1.7	$(2 \times 0.7) \times 1.7$ = 2.38
5	2011-05-07	u5	r1	t1, t2	0.7	1	$(2 \times 0.7) \times 1$ = 1.4

분석을 위해, 제 4.1절에서는 폭소노미 모의 데이터 셋에 대한 실험을 수행하고, 제 4.2절에서는 실제 폭소노미 사이트인 딜리셔스 데이터 셋에 이점표 데이터를 추가하여 FSRank와 SPEAR를 비교 분석한다.

4.1 모의 실험 유형 및 데이터 셋

이미 언급한 바와 같이, 본 논문의 목적은 랭킹에서 중요하게 고려해야 할 랭킹 팩터별로 가중치를 설정하고, 이들의 조합에 의해 사이트의 랭킹 정책에 부합하는 랭킹 알고리즘을 디자인할 수 있도록 지원하는 종합적인 랭킹 프레임워크를 설계하는 것이다. 이를 위해, 시맨틱 웹 자원 랭킹 알고리즘이 폭소노미 도메인에도 효과적으로 적용될 수 있는지, 각 랭킹 팩터들이 사용자가 원하는 방향으로 랭킹 결과에 반영되는지를 검토하기 위해 <표 4>와 같은 모의 실험을 수행하였다. FSRank_b는 시맨틱 웹 자원 랭킹 알고리즘이 폭소노미 도메인에도 기본적으로 적용가능한지를 확인하기 위한 것이다. 이것은 다른 랭킹 팩터를 추가했을 때 비교를 위한 베이스라인 역할을 한다. 폭소노미에서는 트리플에 대한 가중치를 계산하는 과정에서 <표 3>의 예에서와 같이, 가중치

행렬의 같은 셀에 기록되어야 하는 가중치가 중복적으로 발생하는 경우가 많다. Park *et al.*[2011]의 연구에서는 다른 도메인을 대상으로 실험했기 때문에 이러한 경우를 예상하지 못했었는데, FSRank_b는 이러한 경우에 같은 셀에 대한 가중치들을 누적 합산하여 최종적인 가중치를 기록한다는 점이 Park *et al.*[2011]의 연구와 다르다. FSRank_d와 FSRank_t는 FSRank_b에 각각 발견자와 추종자, 시간가중치 팩터를 적용한 알고리즘이다. 특정 자원이 등록된 후 시간이 흐름에 따라 이에 대한 추종자의 수는 점점 누적되어 발견자와 추종자 팩터에 의한 가중치는 증가하는 반면에, 오래된 데이터에 대한 시간 가중치는 현저하게 감소할 것이다. 이렇게 두 팩터 간에 상충관계가 있으므로, 각 팩터가 랭킹 결과에 제대로 반영되는 지를 확인하기 위해서는 먼저 각 팩터를 분리해서 실험해야 할 필요성이 있다. FSRank_c는 커뮤니티 분석을 위해 페이지랭크 방식을 폭소노미에 적용한 포크랭크[Hotho *et al.*, 2006]처럼, FSRank_b를 특정 사용자, 자원, 태그 또는 이들의 모임과 밀접한 관련이 있는 사용자, 자원, 태그 그룹을 추출하기 위한 용도로 사용할 수 있는 지를 검토하기 위한 것이다.

모의 데이터 셋은 사용자, 자원, 태그와 이들

<표 4> 랭킹팩터별 타당성 검증을 위한 실험 내용

알고리즘 유형	가중치 설정 내용	특징 및 실험 목적
FSRank_b (b: baseline)	속성가중치 <표 1> 적용	폭소노미 하이퍼그래프를 RDF그래프로 변환하여 Park <i>et al.</i> [2011]의 연구를 폭소노미에 맞게 적용한 기본 형태
FSRank_d (d: discoverer-weighted)	FSRank_b+ 발견자와 추종자	FSRank_b에 발견자와 추종자 개념을 적용했을 때 랭킹 변화를 살펴봄으로써 발견자와 추종자 개념 적용의 영향 및 타당성 검증
FSRank_t (t: time-weighted)	FSRank_b+ 시간 가치	FSRank_b에 타임구간별로 다른 가중치를 적용했을 때 오래된 데이터와 최근 데이터의 랭킹 결과를 비교 분석하여 타임 가중치 적용의 타당성 검증
FSRank_c (c: community detection)	FSRank_b+ 커뮤니티 추출	FSRank_b를 특정 개체와 관련도가 높은 그룹을 추출하기 위한 용도로 적용가능한 지를 검토, 특정 개체에 대한 가중치를 높게 했을 때와 일반적인 때의 랭킹 순위 차이를 분석하여 판단

간의 연결 관계를 나타내는 트리플들로 이루어진다. 각 알고리즘이 폭소노미 도메인의 데이터를 타당하게 랭킹하는 지를 확인하기 위해, 랭킹의 결과를 예상할 수 있도록 데이터 셋을 구성하고, 예상한 대로 결과가 나오는지 살펴볼 것이다. 이렇게 하기 위해 먼저, 각 개체에 개체번호를 할당하였다. 개체번호 형식은 '(클래스이름)(타임구간번호)-(인스턴스번호)'이다. 예를 들어, 'user1-3'은 타임구간 '1'에 속하는 개체번호 '3'인 사용자를 의미한다. 랭킹 결과는 분석하기에 용이하도록 타임구간의 수가 작을수록, 인스턴스번호가 작을수록 상위에 랭킹 되도록 한다. 예를 들어, 같은 타임구간 안에서는 'user1-1', 'user1-2', 'user1-3', ...의 순이거나, 타임구간이 다른 경우에는 최신 타임구간일수록 타임구간 번호를 작게 설정하여, 'user1-1', 'user2-1', 'user3-1', ...의 순서로 정렬 되도록 하는 것이다. 자원이나 태그에 대해서도 이

와 같은 방식의 랭킹 결과가 나오도록 할 것이다.

이를 위해, 같은 타임구간 내에서는 특정 개체의 인스턴스 번호가 작을수록 인스턴스 번호가 작은, 다른 많은 개체들과 연결해주었다. 인스턴스 번호가 클수록 중요도 점수는 낮아져야 하므로, 인스턴스번호가 큰 적은 수의 개체들과 연결했다. 또, 사용자를 전문가(expert)와 일반 사용자(user)로 구분하여 전문가는 임의의 자원을 처음으로 발견하고 태그를 할당하며, 전문가의 인스턴스번호가 작을수록 더 많은 자원들을 발견하고, 더 많은 일반 사용자들이 전문가가 발견한 해당 자원에 태그하도록 했다. 또, 전문가는 한 자원에 세 개의 태그를 달고, 사용자들은 두 개의 태그를 달도록 하되, 전문가가 할당한 세 개의 태그 중 앞의 두 개와 같은 태그를 달도록 디자인하였다. <표 5>에 타임구간 1에 속하는 개체들 간에 설정된 링크 관계 및 타임 스탬프정보가 요약되어 있다.

<표 5> 타임구간 1에 속하는 모의 데이터 셋 요약

expert(user)	resource	tag	time stamp
expert1-1	resource1-1 ~ resource1-12 (res. 12개)	tag1-1 ~ 1-36	time1-1 ~ time1-12
expert1-2	resource1-1, resource1-13 ~ resource1-22 (res. 11개)	tag1-1 ~ tag1-2, tag1-37 ~ tag1-66	time1-2, time1-1 ~ time1-10
expert1-3	resource1-1, resource1-23 ~ resource1-30 (res. 9개)	tag1-1 ~ tag1-2, tag1-67 ~ tag1-90	time1-3, time1-1 ~ time1-8
expert1-4	resource1-1, resource1-31 ~ resource1-36 (res. 7개)	tag1-1 ~ tag1-2, tag1-91 ~ tag1-108	time1-4, time1-1 ~ time1-6
expert1-5	resource1-1, resource1-37 ~ resource1-40 (res. 5개)	tag1-1 ~ tag1-2, tag1-109 ~ tag1-120	time1-5, time1-1 ~ time1-4
user1-1 ~ user1-40(user 40명)	resource1-1(res. 1개)	tag1-1 ~ tag1-2	time1-2 ~ time1-41
user1-41 ~ user1-79(user 39명)	resource1-2(res. 1개)	tag1-4 ~ tag1-5	time1-3 ~ time1-41
user1-80 ~ user1-117(user 38명)	resource1-3(res. 1개)	tag1-7 ~ tag1-8	time1-4 ~ time1-41
user1-118 ~ user1-154(user 37명)	resource1-4(res. 1개)	tag1-10 ~ tag1-11	time1-5 ~ time1-41
:	:	:	:
user1-650 ~ user1-667(user 18명)	resource1-23(res. 1개)	tag1-67 ~ tag1-68	time1-2 ~ time1-19
:	:	:	:
user1-701 ~ user1-715(user 15명)	resource1-26(res. 1개)	tag1-76 ~ tag1-77	time1-5 ~ time1-19
:	:	:	:
user1-820(user 1명)	resource1-40(res. 1개)	tag1-118 ~ tag1-119	time1-5

타임 스탬프는 제 4.1.3절의 시간 가중치 적용을 위한 것으로 태깅을 한 시점을 나타낸다.

그런데, FSRank_b와 FSRank_d에서는 시간 가중치를 고려하지 않지만, 시간가중치를 고려하는 FSRank_t를 FSRank_b, FSRank_d와 같은 데이터 셋에 대해 비교하기 위해, <표 5>의 모의 데이터 셋을 확장하여 이와 똑같은 구조의 데이터 셋을 타임구간 2에 대해서도 구성하였다. 타임구간 2에 속하는 데이터의 개체번호의 타임구간번호는 '2'이므로, <표 5>에서 모든 개체번호의 타임구간번호를 2로 바꾼 똑 같은 구조의 데이터 셋이 하나 더 있다고 생각하면 된다. 결과적으로 타임구간 1과 2를 포함하는 전체 모의 데이터 셋에 포함된 사용자는 1650명, 자원은 80개, 태그는 240개이고, 이들 간에 연결관계를 나타내는 트리플은 10,608개이다.

4.1.1 기본적인 상호작용성 개념 적용 가능성 검증: FSRank_b

FSRank_b에서는 타임구간 1과 2를 포함하는 전체 모의 데이터 셋에 대해 <그림 3> 우측이나 <표 1>의 속성 가중치를 적용하였다. 여기에서는 타임구간 1과 2에 시간 가치를 적용하지 않았음에 주의하기 바란다. FSRank_b 랭킹 결과, 사용자와 자원은 각각 expert1-1, expert1-2, expert1-3, ...와 resource1-1, resource1-2, resource1-3, ...순으로 인스턴스번호 순과 완벽하게 일치했으며, 같은 자원에 태깅을 한 사용자들의 점수는 동일하게 계산되었다. 또, <표 6>에서 살펴볼 수 있는 바와 같이, expert1-1과 expert2-1, expert1-2와 expert2-2, resource1-1과 resource2-1, resource1-2과 resource2-2, ...등 타임구간 1과 2에서 서로 대

<표 6> FSRank_b의 상위 20위

rank	expert(user)
1	expert1-1
1	expert2-1
3	expert1-2
3	expert2-2
5	expert1-3
5	expert2-3
7	expert1-4
7	expert2-4
9	expert1-5
9	expert2-5
11	user1-1
11	user2-1
11	user1-2
11	user2-2
11	user1-3
11	user2-3
11	user1-4
11	user2-4
11	user1-5
11	user2-5

rank	resource
1	resource1-1
1	resource2-1
3	resource1-2
3	resource2-2
5	resource1-3
5	resource2-3
7	resource1-4
7	resource2-4
9	resource1-5
9	resource2-5
11	resource1-6
11	resource2-6
13	resource1-7
13	resource2-7
15	resource1-8
15	resource2-8
17	resource1-9
17	resource2-9
19	resource1-10
19	resource2-10

rank	tag
1	tag1-1
1	tag2-1
1	tag1-2
1	tag2-2
4	tag1-3
4	tag2-3
7	tag1-4
7	tag2-4
7	tag1-5
7	tag2-5
11	tag1-7
11	tag2-7
11	tag1-8
11	tag2-8
15	tag1-10
15	tag2-10
15	tag1-11
15	tag2-11
19	tag1-13
19	tag2-13

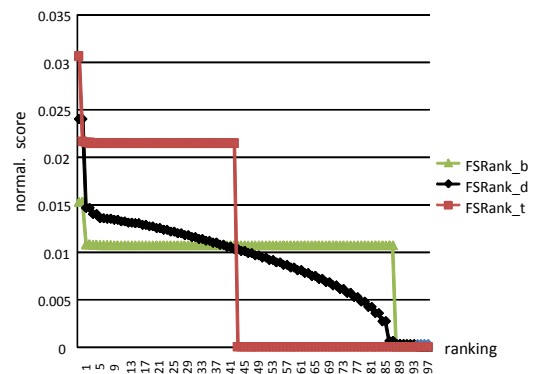
칭 관계에 있는 모든 쌍의 자원들이 서로 같은 점수로 계산되었다. 이것은 폭소노미 도메인에도 상호작용성 개념이 적용될 수 있음을 의미한다. user 1-1부터 user1-5까지는 <표 5>에서 살펴볼 수 있는 것처럼, 모두 resource1-1에 tag1-1, tag1-2를 설정했기 때문에 점수가 서로 같고, user2-1부터 user2-5까지도 같은 구조이기 때문에 즉, resource 2-1에 tag2-1, tag2-2를 설정했기 때문에 점수가 같다. 그리고 FSRank_b는 타임구간 1과 2를 구별하지 않기 때문에 이 10명의 사용자 순위가 모두 11위로 같은 것이다. 태그의 경우에도 쌍을 이루는 현상은 같으며, 하나의 자원에 전문가들은 3개씩, 사용자들은 전문가가 할당된 3개 중 앞의 태그 2개씩을 달도록 데이터를 구성하였으므로, 3의 배수인 인스턴스번호를 갖는 태그들의 점수가 예상한 바와 같이 낮게 나왔다. 'tag1-3'이나 'tag2-3'이 높게 나온 것은 태그의 중요도 계산에 자원의 중요도가 많은 영향을 미치는데, 이 태그가 할당된 'resource1-1'이나 'resource2-1'의 점수가 다른 자원들의 점수 보다 훨씬 높아서인 것으로 분석된다.

이 외에도 특정 개체와 연결된 링크를 삭제하거나 새로운 링크를 추가하였을 때, 해당 개체의 순위가 내려가거나 올라가는 등 데이터의 변화가 랭킹 결과에 타당하게 반영되는 것을 확인하였다. 그런데, 이러한 예는 지면 관계상 따로 소개하지는 않는다.

4.1.2 발견자와 추종자 개념 적용 및 결과 분석: FSRank_d

FSRank_d도 FSRank_b에서와 같이 타임구간 1과 2에 대해 같은 속성 가중치를 적용하였다. 그러므로 여기에서의 추종자 가중치는 추종자에 해당하는(user u, tagged, resource r) 트리플의 주어부 속성 가중치의 합에 1을 더해 계산된다. 타임구간 1과 2는 링크 구조가 같고 이에 따라 서로 대응되는 사용자들의 추종자 구조도 같으

로, 대응관계에 있는 사용자들의 점수가 서로 같았다. 사용자 점수의 총합이 1이 되도록 정규화했을 때, <그림 7>에서와 같이, 사용자 상위 100위의 점수를 살펴보면, FSRank_b에서는 사용자 11위부터 90위까지의 점수가 같은데 반하여, FSRank_d에 의한 결과에서는 타임구간 1과 2에 대해 대칭인 사용자만 한 쌍씩 서로 동점이면서 서로 다른 쌍에 대해서는 상위와 하위의 점수 차이가 뚜렷해짐을 관찰할 수 있다. 이것은 FSRank_b에서는 resource1-1에 같은 태그들을 단 user1-1부터 user 1-40과, 이와 같은 구조로 resource2-1에 같은 태그들을 단 user2-1부터 user2-40이 모두 같은 점수를 받았고(<표 5> 참조), FSRank_d에서는 같은 자원에 같은 태그를 달았어도 액션 시점에 따라 달라지는 각 사용자의 추종자들을 반영했기 때문이다. 전체적으로 FSRank_d는 FSRank_b보다, 상위에 랭킹된 사용자들의 점수는 더 높고 하위에 랭킹된 사용자들의 점수는 더 낮은 분포를 보인다. 사용자들의 구체적인 점수와 순위는 부록의 <부표>에 요약되어 있다.



<그림 7> FSRank_b와 FSRank_d의 사용자 순위에 대한 점수 비교

4.1.3 시간가중치 적용 및 결과 분석: FSRank_t

FSRank_t에서는 시간 가중치의 영향력을 살펴

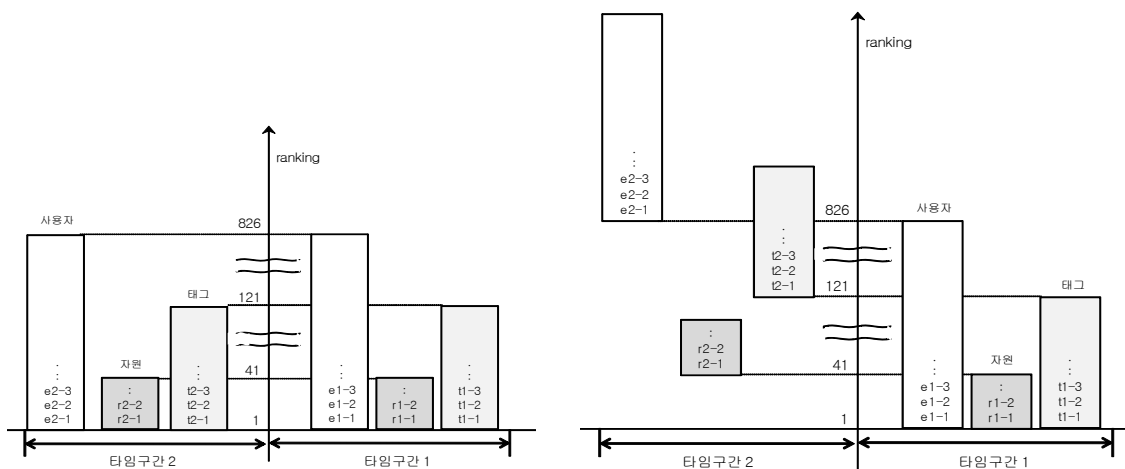
보기 위해, 타임구간 2의 속성가중치를 타임구간 1의 1/3배로 조정하여 적용하였다. <부표>를 살펴보면, FSRank_t의 경우에 상위 사용자의 점수가 상대적으로 제일 높고 하위 사용자의 점수는 제일 낮은 분포를 이루고 있다. 이것은 FSRank_b나 FSRank_d는 타임구간 1과 2에 해당하는 트리플들을 시간 가중치를 고려하지 않고 모두 같은 타임구간에 속하는 것처럼 계산한 반면에 FSRank_t는 타임구간 2에 속하는 트리플들에는 1/3배의 가중치를 적용했기 때문일 것이다. 앞의 <그림 7>에서는 사용자들의 인스턴스번호는 알 수 없지만, <부표>과 비교 해보면, FSRank_b나 FSRank_d는 타임구간 1과 2에 속하는 사용자들을 쌍으로 리스트하고 있고, FSRank_t는 타임구간 1에 속하는 사용자들을 상위에 랭킹하고 있다. FSRank_b의 동점자들이 FSRank_t에서는 타임구간에 따라 상하로 양분됨을 볼 수 있다.

<그림 8>은 사용자를 포함한 자원, 태그의 경우에도 Park *et al.*[2011]을 폭소노미에 적용한 FSRank_b는 타임구간 1과 2의 대응 인스턴스들을 같은 순위로 랭킹하고 있는 반면에, FSRank_t는 타임구간 1에 속하는 인스턴스를 타임구간 2의 대응 인스턴스보다 높게 랭킹하고 있음을 보여

준다. 타임구간 2 내에서는 타임구간 1에서의 상대적인 랭킹 순위가 유지되고 있다. <그림 8>에서는 타임구간 1에 속하는 모든 인스턴스들의 순위가 타임구간 2에 속하는 인스턴스들의 순위 보다 높게 나타나고 있는데, 타임구간 2에 대한 가중치를 1/3배가 아닌 2/3배 등 현재보다 더 높게 설정한다면, 이렇게 이분되는 모습이 아닐 수도 있다. 즉, 타임구간 2의 상위 인스턴스들이 타임구간 1의 하위 인스턴스들 보다 더 높은 점수를 받을 수도 있다는 것이다. 결과적으로 이 예에서는 가장 단순한 계단형 형태의 시간 가중치 함수를 가정하였지만, 시간 가치의 감소율은 시간 가중치 함수의 형태로 조정할 수 있고, FSRank_b에 시간 가중치를 적용하면 오래된 데이터들의 중요도를 체계적으로 떨어뜨릴 수 있음을 보여 준다.

4.1.2 폭소노미 커뮤니티 분석을 위한 적용 가능성: FSRank_c

커뮤니티 분석을 위해서는 링크 연결 구조를 좀 더 구체적으로 파악해야 하므로, 분석의 용이성을 위해 전체 데이터 셋이 아닌 <표 5>의 타임구



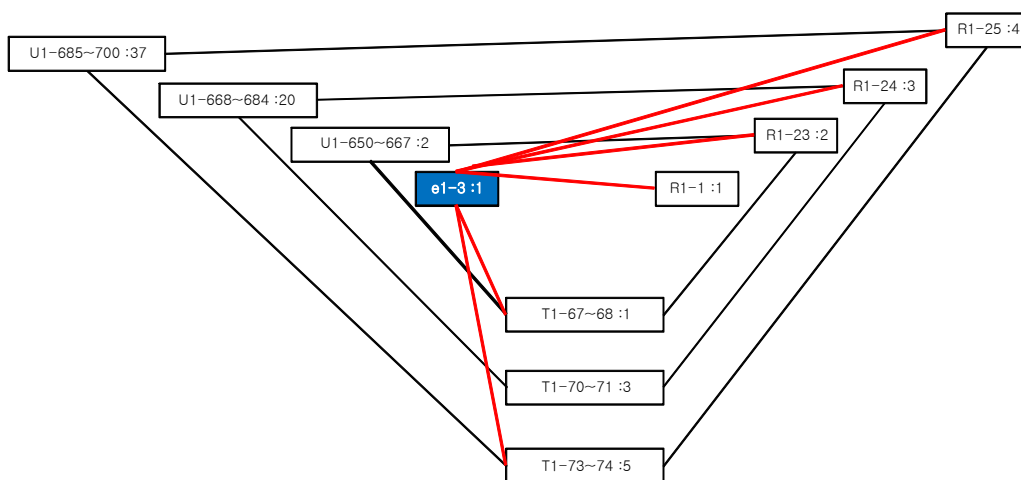
<그림 8> FSRank_b (좌)와 FSRank_t (우) 랭킹 비교

간 1에 해당하는 데이터 셋에 대해 랭킹 연산을 수행하였다. FSRank_c도 포크랭크[Hotho *et al.*, 2006]와 같이, 폭소노미에서의 커뮤니티 분석을 위해 적용 가능한 지를 살펴보기 위한 예로, 'expert1-3'의 커뮤니티가 합리적으로 추출되는지를 분석해 보았다. 먼저, 'expert1-3'에 대한 트리플의 주어부와 목적부 가중치를 기본 가중치의 30배를 하여 점수 벡터를 계산하고 모든 개체들의 점수 합이 1이 되도록 정규화하였다. 설명의 편의를 위하여 이것을 FSRank_p라 하자. 그리고 FSRank_b에 의해 얻은 점수 벡터를 같은 방법으로 정규화한 다음, FSRank_p와 FSRank_b의 차이를 계산하여 FSRank_c 랭킹 벡터를 구하였다. <그림 9>는 사용자, 자원, 태그의 상위 랭킹에 해당하는 인스턴스번호와 이들의 순위(':' 다음의 수) 및 링크 연결관계를 보여준다. 사용자 중에서는 'expert1-3'이 가장 높은 점수를 받았으며, 'expert1-3'과 같이 'resource1-23'에 같은 태그 'tag1-67' ~ 'tag1-68'을 할당한 추종자들 'user1-650' ~ 'user1-667'이 다음으로 높은 점수를 받았고, 'resource1-24'에 'tag1-70' ~ 'tag1-71'을 할당한 추종자들 'user1-668' ~ 'user1-684'이 다음을 이었다(<표 5> 참조). 그 다음 순위도 비슷한 경우다. 이렇게 'expert1-3'

과 같은 자원 같은 태그에 연결된 사용자들은 'expert1-3'과 함께 랭킹 순위가 놀랍게 상승하였음을 확인할 수 있다. 이들과 연결된 자원 및 태그 그룹의 순위도 현격한 차이를 보인다. 이렇게 'expert1-3'과 밀접한 관계가 있는 개체들은 점수가 올라가고 다른 개체들의 점수는 상대적으로 감소하여 'expert1-3'의 커뮤니티 추출이 가능하다. 다른 개체나 개체군들에 대해서도 비슷한 실험을 반복하여, 데이터 구조와 일치하는 결과를 얻었지만 이에 대한 서술은 생략하기로 한다. 결과적으로, 이렇게 커뮤니티를 추출하는 방법을 응용하면, 마케팅 타겟 그룹 선정이나 개인화 인터페이스 등 다양한 지원 도구로 활용할 수 있을 것으로 생각된다.

4.2 실제 사이트 랭킹 적용 및 결과

제 4.1절의 모의 실험은 폭소노미 랭킹 팩터들이 랭킹 결과에 어떻게 반영되는지를 살펴보고, 타당성을 검증하며, 종합적인 랭킹 프레임워크에서 가중치 조정과정에 대한 이해를 돕기 위한 것이었다. 본 절에서는 각 랭킹 팩터들이 결합된 FSRank의 실제 구현가능성과 함께 SPEAR와의 비교분



<그림 9> FSRank_c에 의한 'expert1-3'의 커뮤니티 추출

석을 위한 실험을 수행한다.

<표 7> 실제 데이터에 삽입된 이정표 데이터의 사용자 유형

4.2.1 최종 데이터 셋 및 실험 방법

실제 데이터 셋은 연구 목적을 위해 오픈되어 있는 2007년 7월부터 2007년 12월까지의 딜리셔스 데이터를 사용하였다. SPEAR와의 비교를 위해 수많은 태그들을 먼저 K-평균군집화(K-mean clustering) 방법으로 클러스터링 한 후, 이 중 (mac, apple, osx, ipod, itunes) 클러스터를 선택하여 이 태그들과 연결되어 있는 서브 그래프를 추출하였다. 그런데, 이러한 실제 그래프 데이터는 개체들이 불규칙적으로 복잡하게 연결되어 있어, 어떻게 랭킹하는 것이 정답인지를 정확히 알 수 없다. 전문가들이 네트워크로 연결된 엄청난 양의 과거 데이터들을 살펴보고 평가해보는 것도 가능한 하겠지만, 이에 소요되는 노력 비용에 비해 최종적으로 얻게 되는 랭킹 결과에 대한 신뢰도는 매우 낮을 것으로 판단하였다. 결국 랭킹 성능을 비교 분석할 만한 그라운드 트루쓰(ground truth)가 없으므로, 랭킹 정답을 알 수 있는 이정표 데이터를 실제 데이터 셋에 삽입한 후, 실제 데이터와 혼합되어 출력되는 이정표 데이터의 랭킹 결과를 살펴봄으로써 실제 데이터의 랭킹 결과를 간접적으로 평가해보는 방법을 사용하였다.

이정표 데이터 <표 7>의 사용자들은 과거와 현재에 모두 적극적으로 활동하는 그룹 A, 과거에는 소극적이었다가 현재에 적극적인 그룹 B, 과거에는 적극적이었지만 현재는 소극적인 그룹 C, 과거와 현재 모두 소극적인 그룹 D로 분류되어 있고, 각 그룹 안에서도 각 사용자의 인스턴스 번호 순으로 우열이 정해지도록 URL 등록 수와 각 URL에 대한 추종자 수를 조정하였다. 그리고 그룹 B와 C에 속한 사용자들은 과거와 현재를 서로 대칭적으로 구성하였다. 제 4.1절의 모의 데이터와 다른 점은 <그림 5>와 같이 발견자와 추종자 그리고 시간 가중치 개념을 함께 적용할 경우의 랭킹 결과를 확인할 수 있다는 것이다.

그룹 유형 및 사용자		과거: 타임구간 3 (2007년 7~8월)		현재: 타임구간 1 (2007년 11~12월)		연결 태그
		URL 등록수	추종자 수	URL 등록수	추종자 수	
Group A	userA-1	29	49	30	50	tag1
	userA-2	30	50	29	49	tag1
	userA-3	28	48	29	49	tag1
	userA-4	29	49	28	48	tag1
	userA-5	27	47	28	48	tag1
	userA-6	28	48	27	47	tag1
	userA-7	26	46	27	47	tag1
	userA-8	27	47	26	46	tag1
	userA-9	25	45	26	46	tag1
	userA-10	26	46	25	45	tag1
Group B	userB-1	5	10	25	45	tag2
	userB-2	5	9	25	44	tag2
	userB-3	5	8	25	43	tag2
	userB-4	5	7	25	42	tag2
	userB-5	5	6	25	41	tag2
	userB-6	4	6	25	40	tag2
	userB-7	3	6	25	39	tag2
	userB-8	3	5	25	38	tag2
	userB-9	2	5	25	37	tag2
	userB-10	2	4	25	36	tag2
Group C	userC-1	25	45	5	10	tag3
	userC-2	25	44	5	9	tag3
	userC-3	25	43	5	8	tag3
	userC-4	25	42	5	7	tag3
	userC-5	25	41	5	6	tag3
	userC-6	25	40	4	6	tag3
	userC-7	25	39	3	6	tag3
	userC-8	25	38	3	5	tag3
	userC-9	25	37	2	5	tag3
	userC-10	25	36	2	4	tag3
Group D	userD-1	2	4	2	4	tag4
	userD-2	2	3	2	4	tag4
	userD-3	2	4	2	3	tag4
	userD-4	2	3	2	3	tag4
	userD-5	2	2	2	3	tag4
	userD-6	2	3	2	2	tag4
	userD-7	2	2	2	2	tag4
	userD-8	1	2	2	2	tag4
	userD-9	1	1	2	2	tag4
	userD-10	1	1	1	2	tag4

그런데, 실제 데이터가 이정표 데이터에 서로 다른 영향을 미친다면, 이정표 데이터의 랭킹 결과가 원래 예측했던 것과 달라질 것이므로, 이정표의 역할을 할 수 없을 것이다. 그래서 간편한 방법으로 <그림 10>에서와 같이 이정표 데이터에 있는 모든 사용자들을 실제 데이터의 같은 날, 같은 URL에, 같은 태그를 할당한 적이 있는 것으로 하여 실제 데이터 그래프에 연결해 주었다. 예를 들어, <그림 10>에서 이정표 사용자인 'user A-1'은 '2007년 11월 7일'에 실제 데이터에 있는 URL 중의 하나인 'http://chris.pirillo.com/2007/11/06/top-100-mac-apps/'에 'mac'이라는 태그를 할당한 적이 있다. 이렇게 구성된 최종 데이터 셋에 대해 랭킹 행렬 연산을 하게 되면, 실제 데이터에 포함된 개체들 사이사이에 이정표 데이터의 개체들이 섞여 나오게 될 것이다. 실제 데이터의 폭소노미 수는 918,458개이고, 이정표 데이터의 폭소노미 수는 34,898개이다. <그림 10>은

이러한 최종 폭소노미 데이터 셋의 일부분을 보여준다.

시간 가중치는 <표 2>와 같은 방식으로, 2007년 11월~12월 데이터에는 속성가중치 그대로, 2007년 9월~10월 데이터에는 속성가중치의 2/3를, 2007년 7월~8월 데이터에는 속성가중치의 1/3을 곱하여 반영하였다. 최종 데이터 셋의 각 개체 수와 이에 대한 가중치 행렬 사이즈는 <표 8>과 같고, 전체 폭소노미 수는 953,356개이며, 이에 대한 트리플 수는 이것의 3배인 2,860,068개이다.

4.2.2 결과 분석 및 알고리즘 비교

FSRank와 SPEAR 랭킹의 가장 뚜렷한 차이는 과거와 현재 타임구간에 대한 시간 가치의 반영 여부에서 비롯된다. <그림 11>에서 가로축은 사용자 순위를, 세로축은 137,971명의 사용자 점수의 합이 100이 되도록 정규화한 점수를 나타낸다. FSRank는 이정표 데이터 A, B, C, D 그룹의 사용

	time	user	URL	tag
실제데이터 일부	2007-11-07	b9600c6387d199833ea153579c318f6a	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	mac
	2007-11-07	b9600c6387d199833ea153579c318f6a	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	apple
	2007-11-07	b9600c6387d199833ea153579c318f6a	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	ipod
	2007-11-21	611458570aa29f2994a5107990df1d32	http://www.usingmac.com/2007/11/21/mac-os-x-leopard-200-productivity-booster-hotkeys	mac
	2007-11-21	611458570aa29f2994a5107990df1d32	http://www.usingmac.com/2007/11/21/mac-os-x-leopard-200-productivity-booster-hotkeys	osx
실제데이터와 이정표데이터 연결부분 일부	2007-11-07	userA-1	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	mac
	2007-11-07	userA-2	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	mac
	2007-11-07	userA-3	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	mac
	2007-11-07	userA-4	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	mac
	2007-11-07	userA-5	http://chris.pirillo.com/2007/11/06/top-100-mac-apps/	mac
이정표데이터 일부	2007-11-01	userA-1	resource1-1	tag1
	2007-11-02	user1-1-1	resource1-1	tag1
	2007-11-03	user1-1-2	resource1-1	tag1
	2007-11-04	user1-1-3	resource1-1	tag1
	2007-11-05	user1-1-4	resource1-1	tag1

<그림 10> 실제 데이터와 이정표 데이터로 이루어진 최종 데이터 셋 일부

<표 8> 항목별 개체 수 및 가중치 행렬 사이즈

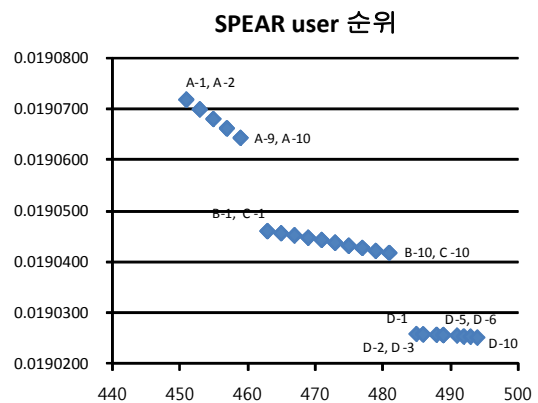
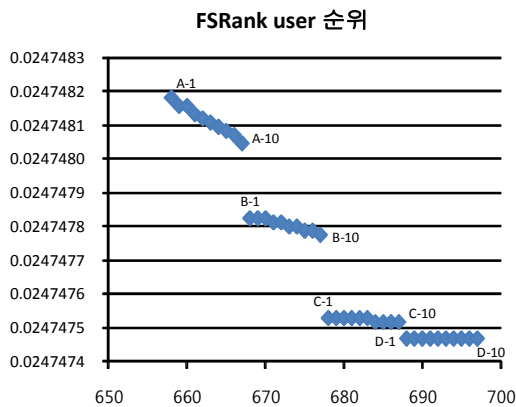
항목	전체 데이터	실제 데이터	이정표 데이터	(실제/이정표) 비율
user	137,971	137,931	40	3448.3
URL	207,931	206,767	1,164	177.6
tag	9	5	4	1.3
합계 및 가중치행렬 사이즈	345,911 (가중치행렬사이즈)	344,703	1,208	285.4

자들을 순서대로 랭킹하고 있다. 반면에 SPEAR는 A-1과 A-2, A-3와 A-4 또는 D-2와 D-3 등과 같이 <표 7>의 같은 그룹 내에서 대칭구조를 가지고 있는 사용자들의 순위를 같게 랭킹하고 있으며, B-1과 C-1, B-2와 C-2 등과 같이 과거에는 소극적이었다가 현재에 적극적인 B그룹과 과거에는 적극적이었지만 현재에는 소극적인 C그룹을 구별하지 못하고 있다.

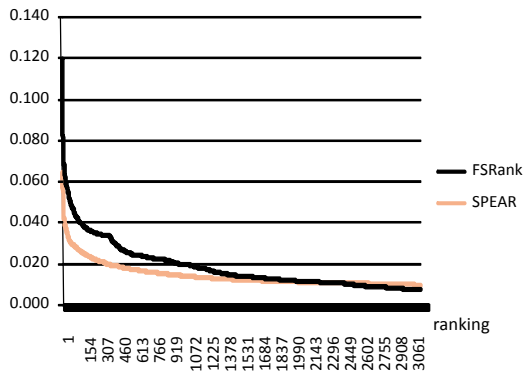
이와 같이, 이정표 데이터에 포함된 개체들의 랭킹 결과를 살펴본 결과, FSRank는 시간 가치를 반영하여 제대로 랭킹하고 있는 반면에, SPEAR는 개체들의 연결구조에 따라 순위 상하를 가려 내지 못하는 경우가 발생한다는 것을 확인하였다. <그림 11>에서 가로축을 보면, 두 알고리즘에 의한 랭킹 순위의 범위가 다르다. 시간 가치를 반영하는 FSRank가 더 낮은 순위 범위에 이정표 데이터를 리스트하고 있는데, 이것은 전체 데이

터 셋 안에 이정표 데이터에 비해 상대적으로 최신에 이루어진 다른 데이터들이 많다는 것을 의미한다고 볼 수 있다. 참고로 상위 사용자 3,200 등까지의 점수 분포를 비교해보면, <그림 12>와 같다. 두 알고리즘 모두 상위로 갈수록 점수 차이가 뚜렷해지는 형태를 보인다. 그리고 FSRank가 상위 사용자들의 점수를 훨씬 더 높게(1등 점수: $FSRank \approx 0.12$, $SPEAR \approx 0.065$), 상대적으로 하위 사용자들은 더 낮게 평가하고 있음을 볼 수 있다.

이제 FSRank의 랭킹 결과가 SPEAR보다 타당하다는 사실에 근거하여, FSRank와 SPEAR의 랭킹 결과가 전체 데이터 셋에 대해서는 어느 정도 차이가 나는지를 살펴보기 위해, <그림 13>과 같은 랭킹 유사도를 측정해 보았다. <그림 13>에서 가로축은 순위(k)이고 세로축은 랭킹 유사도로, 전체 데이터에 대한 FSRank와 SPEAR의 랭킹 유사도를 보여준다.



<그림 11> FSRank와 SPEAR의 이정표 데이터 사용자 랭킹(정규화: 전체사용자점수의 합 = 100)



<그림 12> 사용자 랭킹 점수 분포(상위 3,200등까지, 정규화: 전체사용자점수의 합 = 100)

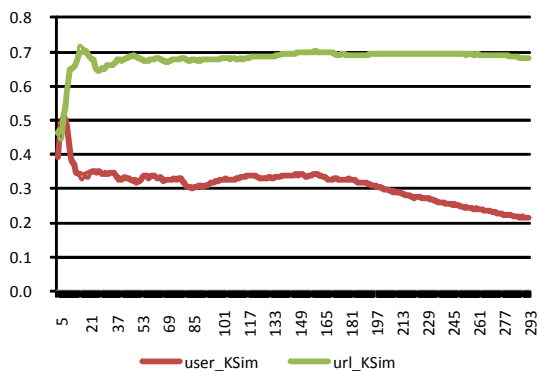
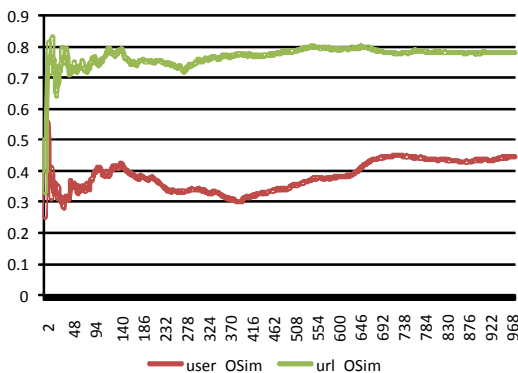
OSim[Haveliwala, 2003]은 $OSim(\tau_1, \tau_2) = \frac{|R_1 \cap R_2|}{k}$ 로 정의되며, 두 랭킹 τ_1 과 τ_2 의 각 상위 k 랭킹에 해당하는 개체집합 R_1 과 R_2 의 교집합의 원소 수를 나타낸다. KSim[Kendall, 1938; Haveliwala, 2003]은

$$KSim(\tau_1, \tau_2) = \frac{\left| \left\{ (r_u, r_v) : \tau_1, \tau_2 \text{ agree on order of } (r_u, r_v), r_u \neq r_v \right\} \right|}{|R_{\tau_1 \cup \tau_2}| \times (|R_{\tau_1 \cup \tau_2}| - 1)}$$

로 정의되며, 상위 k 랭킹에 대한 합집합 안에서 상대적 순위가 같은 서로 다른 개체의 쌍의 수에 따라 달라진다. 즉, τ_1 랭킹에서의 순위 상하가 τ_2 에서의 순위 상하와 같은, 서로 다른 개체의 쌍의 수가 많을수록 1에 가까운 값을 갖는다. FSRank와 SPEAR의 랭킹 결과는 기본적으로 서로 같지 않

고, 사용자 순위의 유사도는 OSim과 KSim 모두에 대해 URL 자원 순위의 유사도보다 훨씬 낮다. 이것은 사용자의 점수를 계산할 때에는 발견자와 추종자 개념도 반영되는데, FSRank와 SPEAR의 반영 방식이 서로 달라서인 것으로 판단된다.

<표 9>는 지금까지 설명된 내용을 바탕으로 FSRank와 SPEAR를 비교하여 정리한 것이다. 본 실험에서 트리플 파일과 트리플에 포함된 서로 다른 인스턴스 파일을 생성하고, 인스턴스 파일과 가중치 파일을 로딩하여 사이즈 345,911의 정방 가중치 행렬을 생성하고, 500회의 이터레이션을 통하여 랭킹 점수 파일을 생성할 때까지 소요되는 시간을 비교해보면, FSRank가 약 1분 44초인데 비해 SPEAR는 약 3시간 2분 21초였다. 시스템 환경은 Intel(R) Xeon(R) E5430(2.66GHz, 4 core 64bit) CPU와 RAM 8GB, CentOS release 5.5(64bit) OS와 JVM 1.6.0_17(64bit)이다. 최대 메모리 사용량도 FSRank가 약 1,925MB인데 비해 SPEAR는 약 6,703MB로 큰 차이가 있었다. 연산 방법이나 메모리 관리 방식에 따라 차이의 폭이 달라질 수는 있겠지만, FSRank는 처음부터 하나의 가중치 행렬을 만드는데 비해, SPEAR는 주어부와 목적부 행렬을 따로 만들고 이 두 정방행렬을 곱하여 최종 가중치행렬을 만드는 이터레이션 이전 단계의 준비과정이 필요하기 때문에 두 방법의 근본적인 차이는 해소되기 힘들 것으로



<그림 13> FSRank와 SPEAR에 의한 사용자/URL 순위의 OSim과 KSim

<표 9> FSRank와 SPEAR의 랭킹연산 및 가중치 설정방식 비교

대분류	소분류	FSRank	SPEAR
랭킹 연산	랭킹 패러다임	링크 방향과 무관한 시맨틱 상호작용성	링크 방향에 의한 투표성 개념
	랭킹 대상	다양한 종류의 이질적인 개체들을 동시에 랭킹 (사용자, 자원, 태그)	세 가지 이상의 이질적인 개체들을 동시에 랭킹 불가능 (사용자, 자원)
	초기 가중치 행렬 구성	하나의 가중치 행렬에 의해 하나의 중요도 점수 계산	주어부와 목적부 행렬을 따로 구성하여 각각을 곱하는 순서에 따라 권위점수와 허브 점수가 계산됨 (사용자는 허브점수, 자원은 권위점수)
	연산 효율성	이터레이션의 횟수는 연산시간에 크게 영향을 주지 않음(본 논문 제4.2절 실험의 경우 이터레이션 1회에 약 0.05초 소요됨)	이터레이션 전 단계에서 주어부와목적부 행렬을 곱하기 위한 추가비용 발생
가중치 설정 인터페이스	기본 속성 가중치 설정 방식	개체 중요도에 대한 트리플 속성의 비중에 따라 사이트 관리자가 설정	개체간 링크가 있으면 1, 없으면 0으로 일률적으로 설정
	발견자와 추종자 가중치	트리플의 액션 시점을 고려한 속성 가중치를 더하여 계산, 모든 추종자에 대해 적용할 수도 있고, 10명 이상의 추종자가 모인 자원에 대해 5명까지만 추종자 점수를 주는 등 랭킹 정책 반영 가능	추종자 총 수의 양의 제곱근
	시간 가중치	주/월별 단순 계단형, 일별 곡선형 등 다양한 시간 가중치 함수 적용 가능	고려하지 않음, 추종자가 누적되어 오래된 데이터의 영향력이 커질 수 있음
	커뮤니티 분석	특정 개체와 연관성이 높은 그룹 추출 가능	고려하지 않음

보인다. SPEAR의 허브점수를 구할 때 두 정방행렬을 곱하는 과정에만 소요된 시간은 57분 3초이며, 권위점수의 경우에도 비슷한 시간이 걸렸다.

V. 결론 및 향후 연구방향

앞에서 폭소노미 시스템에는 과거 데이터 보다는 최신 데이터를 더 높게 평가하고, 사용자의 전문성을 평가할 필요가 있으며, 커뮤니티 분석 등의 랭킹 니즈들이 있고, FSRank가 이러한 랭킹 니즈를 잘 반영할 수 있음을 살펴보았다. FSRank는 SPEAR의 핵심인 발견자와 추종자 개념을 시간 가치 적용을 통해 더욱 탄력적으로 조정할 수 있으며, 기본 속성가중치에도 시간 가치를 반영

함으로써 인기 있는 최신 데이터를 상위에 랭킹할 수 있다. SPEAR는 HITS의 링크 방향에 의한 투표성 개념을 근간으로 하기 때문에 도메인의 클래스와 속성이 2가지 이상일 때에는 유연성이 떨어지는 단점이 있다. 폭소노미 도메인 구성은 IT기술의 변화에 따라 사용자와 URL 자원 사이의 관계인 ‘트위터로 보내다’나 사용자와 사용자 사이의 관계인 ‘친구로 등록하다’ 등 랭킹 점수계산에 고려할만한 새로운 속성들을 추가함으로써 더욱 복잡해질 수 있다. SPEAR는 가중치행렬연산에 있어서도 이터레이션 전 단계에서 두 개의 행렬을 곱해야 하는 과정이 추가적으로 소요된다. 결론적으로, 그 동안 페이지랭크와 HITS 이후 여러 분야에서 이들을 적용하는 많은 연구들

이 진행되어왔고 폭소노미에서도 비슷한 연구들이 있었지만, 투표 개념보다는 FSRank와 같이 새로운 상호작용성 개념으로 접근하는 것이 더욱 효과적이라고 할 수 있다.

이상과 같이, 본 논문의 공헌점은 폭소노미 도메인에 대한 상호작용성 기반 랭킹 알고리즘의 적용 가능성을 밝히고, 폭소노미의 랭킹 정책에 따라 랭킹 팩터 별로 조정 가능하고, 기술의 변화에 따라 새로 추가될 만한 속성들도 반영 가능한 종합적인 랭킹 프레임워크를 설계했다는 것이다. 특히, 기본적인 상호작용성 개념 위에 소셜 미디어의 중요한 특성인 시간 가치를 적용할 수 있는 방법을 최초로 제시했으며, 이 방법은 다른 소셜 미디어 서비스에도 적용 가능할 것으로 보인다.

본 연구를 수행함에 있어 가장 어려웠던 점은 랭킹 결과를 비교 분석하는 방법론을 결정하는 일이었다. 다른 랭킹 관련 연구에서는 실제 해당 도메인에 대한 사용자들의 평가 결과와 논문에서 제안되는 알고리즘의 랭킹 결과를 비교하는 방식으로 접근한 예도 있다. 그런데, 폭소노미 도메인에서는 사용자들의 평가가 가능한, 최근 데이터를 구하는 것이 어려웠고, 연구 목적으로 사용할 수 있는 오픈 데이터는 모두 과거 자료였다. 결국 대부분의 랭킹 이슈가 그렇듯이, 최종 데이터 셋의 랭킹에 대한 그라운드 트루쓰가 없고, 그라운드 트루쓰를 대체할만한 전문가 평가도 합리적인 대안이 될 수 없는 상황이었다. 그래서, 본 논문에서는 랭킹 점수의 우열이 분명하게 예상되는 이정표 개체들을 만들어 이들을 실제 데이터와 혼합하여 랭킹함으로써, 이들을 중심으로 랭킹 결과를 비교하는 방법을 사용하였다. 그런데, 특정 사용자가 특정 태그로 등록한 자원의 유용성이 높다면 이에 대한 추종자들이 많을 것이고, 과거 데이터의 가치는 시간이 지나면서 하락하는 등 전문가의 평가 요소를 랭킹 팩터로 알고리즘에 그대로 반영하고 있으므로, 이 방법이 전문가의 평가에 비해 객관성이 떨어지는 방법은 아닌 것 같다. 이미 모의 데이터에 대한 사전 테

스트를 통해 전문가들이 예측한 결과와 알고리즘의 랭킹 결과가 같음을 확인하였고, 데이터의 규모가 클 때에는 사람의 인식 오차를 줄여주는 면이 있기 때문이다.

끝으로, 본 연구에서 제안한 랭킹 프레임워크를 구현하기 위해 충분히 고려해야 할 사항 및 추후연구 방향은 다음과 같다. 첫째, 랭킹 연산을 위해서는 중요도에 영향을 미치는 속성들의 가중치와 시간 가중치 함수를 합리적으로 조정하는 과정이 필요하며, 랭킹 프로그램은 이들의 변화를 유연하게 반영할 수 있도록 디자인되어야 한다. 둘째, 특정 주제에 대한 전문가나 자원을 검색하는 경우, 해당 태그들과 관련된 서브 네트워크에 대해 랭킹한 결과를 사용할 수도 있고, 전체 네트워크에 대해 랭킹 연산을 실행한 후 해당 주제와 관련된 사용자들의 랭킹 점수를 사용할 수도 있다. 해당 주제와 거리가 먼 개체들의 영향으로 인한 토픽 드리프트(topic drift) 현상 때문에, 첫 번째 방법이 더 낫다는 연구들도 있지만, 두 가지 모두 특정 주제에 속하는 태그 클러스터를 선정하는 방안이 마련되어야 한다. 이것은 다른 HITS 기반 알고리즘의 경우에도 해당되는 문제이다. 태그 클러스터링에 관한 다양한 알고리즘들이 있지만 이 중 어떤 방법이 폭소노미에 가장 적합한지에 대한 논의는 본 논문의 연구 범위를 벗어나 언급하지 않았다. 셋째는 모든 그래프 기반 랭킹 알고리즘의 문제라고 할 수 있는 것으로, 랭킹 점수가 높은 사용자나 자원에 연결된 비활동적인 사용자나 자원이 상대적으로 쉽게 점수를 받을 수 있다는 것이다. 시맨틱 웹 자원 랭킹 알고리즘에서는 속성 가중치를 조정함으로써 이러한 문제점을 최대한 완화할 수는 있지만, 가중치를 설정할 때에, 사용자의 경우에는 전체 자원 중 해당 사용자가 등록한 자원 수 비율을, 자원의 경우에는 전체 사용자 중 해당 자원을 등록한 사용자 수의 비율을 반영하는 등 지속적인 연구를 통해 더욱 향상된 알고리즘으로 완성해볼 수 있겠다.

〈References〉

- [1] Abel, F., Henze, N., and Krause, D., "Analyzing Ranking Algorithms in Folksonomy Systems," Technical Report, L3S Research Center, 2008.
- [2] Abel, F., Henze, N., and Krause, D., "Ranking in Folksonomy Systems: Can Context Help?," *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, 2008, pp. 1429-1430.
- [3] Bao, S., Wu, S., Fei, B., Xue, G., Su, z., and Yu, Y., "Optimizing Web Search Using Social Annotations," WWW, Banff, Alberta, Canada, 2007.
- [4] Brin, S. and Page, L., "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, Vol. 30, No. 1~7, 1998, pp. 107-119.
- [5] Dom, B., Eiron, I., Cozzi, A., and Zhang, Y., "Graph-based Ranking Algorithms for E-mail Expertise Analysis," In Proc. of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, USA, 2003, pp. 42-48.
- [6] Feltovich, P.J., Prietula, M.J., and Ericsson, K.A., "Studies of Expertise from Psychological Perspectives," In the Cambridge Handbook of Expertise and Expert Performance, Cambridge University Press, USA, 2006, pp. 41-68.
- [7] Hammond, T., Hannay, T., Lund, B., and Scott, J., "Social Bookmarking Tools(i): A General Review," *D-Lib Magazine*, Vol. 11, No. 4, 2005.
- [8] Haveliwala, T.H., "Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 4, 2003, pp. 784-796.
- [9] Haveliwala, T.H., *Efficient Computation of PageRank*, Unpublished Manuscript, Stanford University, 1999.
- [10] Hayes, C. and Avesani, P., "Using Tags and Clustering to Identify Topic-relevant Blogs," In International Conference on Weblogs and Social Media, 2007.
- [11] Hayes, C., Avesani, P., and Veeramachaneni, S., "An Analysis of the Use of Tags in a Blog Recommender Systems," In Proceedings of the IJCAI, 2007.
- [12] Hotho, A., Jäschke, R., Schmitz, C., and Stumm, G., "Information Retrieval in Folksonomies: Search and Ranking," In York Sure and John Domingue (Eds.), *The Semantic Web: Research and Applications*, LNAI, Heidelberg, Springer, Vol. 4011, 2006, pp. 411-426.
- [13] Hotho, A., Jäschke, R., Schmitz, C., and Stumm, G., "FolkRank: A Ranking Algorithm for Folksonomies," In Proc. of FGIR 2006.
- [14] Kendall, M.G., "A New Measure of Rank Correlation," *Biometrika*, Vol. 30, No. 1-2, 1938, pp. 81-93.
- [15] Kleinberg, J., "Authoritative Sources in a Hyperlinked Environment," *Journal of the ACM*, Vol. 46, No. 5, 1999, pp. 604-632.
- [16] Klyne, G. and Carroll, J. (Eds.), "Resource Description Framework (RDF): Concepts and Abstract Syntax," W3C Recommendation, 2004.
- [17] Manola, F. and Miller, E. (Eds.), "RDF Primer," W3C Recommendation, 2004.
- [18] Mika, P., "Ontologies are us: A Unified Mo-

- del of Social Networks and Semantics," *Journal of Web Semantics*, Vol. 5, No. 1, 2007, pp. 5-15.
- [19] Noll, M.G. and Meinel, C., "Exploring Social Annotations for Web Document Classification," In Proc. of ACM Symposium on Applied Computing, Fortaleza, Brazil, 2008, pp. 2315-2320.
- [20] Noll, M.G., Yeung, C.A., Gibbins, N., Meinel, C., and Shadbolt, N., "Telling Experts from Spammers: Expertise Ranking in Folksonomies," In SIGIR: Proceedings of the 32nd International ACM SIGIR Conference on Research and development in Information Retrieval NY, USA: ACM, 2009, pp. 612-619.
- [21] Orlicki, J.I., Fierens, P.I., and Alvarez-Hamelin, J.I., "Faceted Ranking in Collaborative Tagging Systems: Efficient Algorithms for Ranking Users Based on a Set of Tags," In INSTICC Press, WEBIST 2009: Proceedings of the 5th International Conference on Web Information Systems and Technologies, Lisboa, Portugal, 2009, pp. 626-633.
- [22] Page, L., Brin, S., Motwani, R., and Winograd, T., *The Page Rank Citation Ranking: Bringing Order to the Web*, Technical Report, Stanford University, 1998.
- [23] Park, H., Rho, S., and Park, J., "A Link-Based Ranking Algorithm for Semantic Web Resources: A Class-Oriented Approach Independent of Link Direction," *Journal of Database Management*, Vol. 22, No. 1, 2011, pp. 1-25.
- [24] Wang, J., Chen, Z., Tao, L., Ma, W.-Y., and Wenxin, L., "Ranking User's Relevance to a Topic through Link Analysis on Web Logs," In WIDM: Proceedings of the 4th International Workshop on Web Information and Data Management, USA, 2002, pp. 49-54.
- [25] Wetzker, R., Zimmermann, C., and Bauckhage, C., "Analyzing Social Bookmarking Systems: A Del.icio.us Cookbook," In Proc. of Mining Social Data Workshop, 2008, pp. 26-30.
- [26] Zhang, J., Ackerman, M.S., and Adamic, L., "Expertise Networks in Online Communities: Structure and Algorithms," In Proc. of WWW Conference, Ban, Canada, 2007, pp. 221-230.
- [27] Zhou, D., Orshanskiy, S.A., Zha, H., and Giles, C.L., "Co-ranking Authors and Documents in a Heterogeneous Network," In Proc. of 7th IEEE International Conference on Data Mining, Washington, USA, 2007, pp. 739-744.

〈부표〉 FSRank_b, FSRank_d, FSRank_t 모의 실험 결과-사용자 랭킹 및 점수

	FSRank_b			FSRank_d			FSRank_t	
	사용자	정규화 점수*		사용자	정규화 점수*		사용자	정규화 점수*
1	expert1-1	0.015331244	1	expert1-1	0.024056752	1	expert1-1	0.030662488
1	expert2-1	0.015331244	1	expert2-1	0.024056752	2	expert1-2	0.021702125
3	expert1-2	0.010851063	3	expert1-2	0.014683186	3	expert1-3	0.021607890
3	expert2-2	0.010851063	3	expert2-2	0.014683186	4	expert1-4	0.021560822
5	expert1-3	0.010803945	5	expert1-3	0.014019669	5	expert1-5	0.021529045
5	expert2-3	0.010803945	5	expert2-3	0.014019669	6	user1-1	0.021479537
7	expert1-4	0.010780411	7	user1-1	0.013587323	6	user1-2	0.021479537
7	expert2-4	0.010780411	7	user2-1	0.013587323	6	user1-3	0.021479537
9	expert1-5	0.010764522	9	expert1-4	0.013540107	6	user1-4	0.021479537
9	expert2-5	0.010764522	9	expert2-4	0.013540107	6	user1-5	0.021479537
11	user1-1	0.010739768	11	user1-2	0.013420033	6	user1-6	0.021479537
11	user2-1	0.010739768	11	user2-2	0.013420033	6	user1-7	0.021479537
11	user1-2	0.010739768	13	user1-3	0.013250526	6	user1-8	0.021479537
11	user2-2	0.010739768	13	user2-3	0.013250526	6	user1-9	0.021479537
11	user1-3	0.010739768	15	expert1-5	0.013120687	6	user1-10	0.021479537
11	user2-3	0.010739768	15	expert2-5	0.013120687	6	user1-11	0.021479537
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1639	user1-816	4.17846E-06	1641	user1-775	7.5297E-07	1641	user2-811	2.29986E-77
1639	user2-816	4.17846E-06	1641	user1-784	7.5297E-07	1641	user2-812	2.29986E-77
1639	user1-817	4.17846E-06	1643	user1-792	6.99529E-07	1641	user2-813	2.29986E-77
1639	user2-817	4.17846E-06	1643	user1-799	6.99529E-07	1641	user2-814	2.29986E-77
1645	user1-818	4.10704E-06	1645	user1-805	6.70429E-07	1645	user2-815	2.25869E-77
1645	user2-818	4.10704E-06	1645	user1-810	6.70429E-07	1645	user2-816	2.25869E-77
1645	user1-819	4.10704E-06	1647	user1-814	6.42254E-07	1645	user2-817	2.25869E-77
1645	user2-819	4.10704E-06	1647	user1-817	6.42254E-07	1648	user2-818	2.22009E-77
1649	user1-820	4.04009E-06	1649	user1-819	6.14925E-07	1648	user2-819	2.22009E-77
1649	user2-820	4.04009E-06	1649	user1-820	6.14925E-07	1650	user2-820	2.1839E-77

주) * 정규화 점수는 사용자들만의 점수 총합이 1이 되도록 정규화한 것임.

◆ About the Authors ◆



Hyunjung Park

Hyunjung Park is a researcher of the Institute of Management Research at Seoul National University. She got a PhD of Management Information System from the Graduate School of Business at Seoul National University. Her doctoral dissertation deals with the ranking problem of Semantic Web resources based on link structure, and her research interests include ontology development and application, knowledge management, U-learning, and social network services. She had an experience working at Korea Telecom, where she has participated in many projects related with the optimization of telecommunication systems and policies.



Sangkyu Rho

Sangkyu Rho is Professor of Information Systems in the Graduate School of Business at Seoul National University. He received his Ph.D. from the University of Minnesota. His research interests include Internet business, ontology development, data mining, ranking, and social networking. He has published papers in such journals as IEEE Transactions on Knowledge and Data Engineering, Information Systems, Annals of Operations Research, and Strategic Management Journal.

Submitted : February 18, 2011
1st revision : April 28, 2011

Accepted : June 28, 2011