DOI: 10.3745/KIPSTD.2011.18D.2.101

스크랩 기능을 지원하는 블로그 공간에서 포스트 랭킹 방안: 알고리즘 및 성능 평가

황 원 석[†] · 도 영 주[††] · 김 상 욱[†††]

요 약

블로그의 사용량이 증가함에 따라 다수의 포스트들이 블로고스피어 내에 작성되고 있으며, 이는 검색에서 웹 서퍼가 양질의 포스트를 찾기 어렵게 하는 문제를 가져왔다. 이로 인하여 포스트 검색에서 랭킹을 부여하기 위한 랭킹 알고리즘의 필요성이 부각되고 있다. 기존에 웹 문서를 위한 다양한 랭킹 알고리즘들이 있었으나, 웹 문서와 포스트의 차이로 인하여 직접 적용하기 어렵다는 문제점이 존재한다. 본 논문에서는 블로거들이 포스트에 남긴 블로그 액션을 이용하여 포스트에 랭킹을 부여하는 방안인 포스트 랭킹 알고리즘들을 제안한다. 그리고 실제 블로그 데이터를 이용하여 포스트 랭킹 알고리즘들의 성능을 분석하고, 이를 바탕으로 블로그에 적합한 포스트 랭킹 알고리즘을 선별한다.

키워드 : 블로고스피어, 포스트 랭킹 알고리즘, 성능 평가

Post Ranking in a Blogosphere with a Scrap Function: Algorithms and Performance Evaluation

Won-Seok Hwang[†] · Young-Joo Do[††] · Sang-Wook Kim[††]

ABSTRACT

According to the increasing use of blogs, a huge number of posts have appeared in a blogosphere. This causes web surfers to face difficulty in finding the quality posts in their search results. As a result, post ranking algorithms are required to help web serfers to effectively search for quality posts. Although there have been various algorithms proposed for web-page ranking, they are not directly applicable to post ranking since posts have their unique features different from those of web pages. In this paper, we propose post ranking algorithms that exploit actions performed by bloggers. We also evaluate the effectiveness of post ranking algorithms by performing extensive experiments using real-world blog data.

Keywords : Blogosphere, Post Ranking Algorithms, Performance Evaluation

1. 서 론

블로그(blog)는 그들의 소유주인 블로거에 의해 관리되는 일종의 개인 웹 사이트이다. 블로그는 포스트(post)들로 구성되어 있으며, 포스트를 관리하기 위한 다양한 기능을 블로거(blogger)에게 제공하고, 이 기능들을 본 논문에서 블로그 액션(blog action)이라 정의한다. 블로그 액션으로는 포스트작성, 덧글달기, 스크랩하기, 포스트엮기 등이 있다. 블로그 액션을 통해 블로거는 포스트에 자신의 의견을 남기거나 다른 블로거와 의견을 나눌 수 있다. 블로그와 포스트, 그리고 그들 사이의 여러 블로그 액션들로 이루어진 사회연결망을 블로고스피어(blogosphere)라고 부른다.

블로그는 자신의 의견이나 생각을 웹에 쉽게 게재할 수 있도록 하는 다양한 기능들을 제공하고 있다. 이는 블로그 사용을 활성화시켜, 다양한 내용을 담은 다수의 포스트들이 지속적으로 생성되도록 하였다. 대부분의 블로그는 다수의 포스트들 중 사용자가 원하는 내용을 담은 포스트를 찾도록 도와주는 검색 기능을 제공하고 있다. 기존의 포스트 검색에서는 사용자가 원하는 키워드인 질의어를 입력 받고 해당 질의어와 매칭되는 포스트를 찾아주는 것만을 고려하였다. 그러나 다수의 포스트가 사용자의 질의어와 매칭되는 경우,

※ 본 연구는 NHN(주)의 지원을 받았습니다. 그러나, 본 논문에서 제시된 의견이나 결론, 또는 권고 등은 온전히 저자(들)의 것이며, 반드시 지원회사의 입장을 대변하는 것은 아닙니다. 또한, 본 연구는 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 부분적인 지원을 받았습니다(No. 2008-0061006). 또한, 본 연구는 지식경제부 및 정보통신산업 진흥원의 'IT융합 고급인력과정 지원사업'의 연구결과로 수행 되었습니다(NIPA-2011-C6150-1101-0001).
† 준 회 원 : 한양대학교 전자통신컴퓨터공학과 박사과정
†† 준 회 원 : 매크로임팩트(주) 연구원
††† 종신회원 : 한양대학교 정보통신대학 정보통신학부 교수
논문접수 : 2010년 12월 2일
수 정 일 : 1차 2011년 1월 5일, 2차 2011년 1월 24일
심사완료 : 2011년 1월 24일

사용자는 키워드 매칭 결과에서 양질의 내용을 담고 있는 포스트를 찾기 위해 결과를 다시 한 번 살펴보아야 하는 어려움이 있다. 이러한 문제를 풍부함의 문제(abundant problem)라 부른다[1]. 이 문제를 해결하기 위하여 키워드 매칭 결과로 나타난 포스트에 랭킹을 부여하는 랭킹 알고리즘이 필요하다. 랭킹 알고리즘은 검색 사용자가 원할 것이라 생각되는 품질이 좋은 포스트에 높은 랭킹을 부여하여, 사람들이 검색 결과의 일부만 확인하고도, 자신이 원하는 포스트를 쉽게 찾을 수 있도록 한다.

기존의 웹 문서 검색에서도 풍부함의 문제가 존재하였고, 이를 해결하기 위해 웹 문서의 랭킹을 결정하는 다양한 알고리즘들이 제안되었다. 특히, 웹 문서와 그들 사이에 존재하는 하이퍼링크를 이용하여 랭킹을 부여하는 알고리즘들을 통틀어 본 논문에서는 웹 문서 랭킹 알고리즘이라 부르기로 한다. 웹 문서를 작성하는 사람은 자신이 만족할만한 품질이거나 정보를 담고 있는 웹 문서를 하이퍼링크를 통해 연결하는 경향이 있다. 따라서 웹 문서 랭킹 알고리즘들은 하이퍼링크를 일종의 추천의 의미로 생각하고 이를 분석하여 각 웹 문서의 랭킹을 계산한다. 웹 문서 랭킹 알고리즘의 예로는 Indegree[2], PageRank[3], HITS[1], SALSA[4] 등이 있다. 또한, 기존의 PageRank와 HITS를 확장한 다음과 같은 다양한 알고리즘들이 제안되었다. 참고문헌 [9][10][11][12][13]은 PageRank를 수정한 알고리즘들이고, 참고문헌 [2][14][15]는 HITS를 수정한 알고리즘들이다. 참고문헌 [16][17]은 PageRank와 HITS를 통합한 알고리즘들을 제안하였다.

포스트 또한 일종의 웹 문서라고 할 수 있기 때문에 포스트의 랭킹을 구하기 위하여 웹 문서 랭킹 알고리즘을 동일하게 적용할 수 있다. 그러나 참고문헌 [5]에서는 하이퍼링크가 블로고스피어 내에 매우 적기 때문에 대부분 포스트의 점수가 측정될 수 없음을 보였다. 하이퍼링크가 부족한 문제를 해결하기 위하여 본 논문에서는 하이퍼링크를 대신하여 블로그 액션을 이용하고자 한다. 블로그 액션은 하이퍼링크와 유사하게 추천의 의미를 담고 있기 때문이다. 특히, 여러 블로그 액션 중 스크랩하기와 포스트엮기는 추천의 의미를 가장 강하게 담고 있는 블로그 액션으로 본 논문에서는 이를 이용하여 포스트에 랭킹을 부여하고자 한다. 그러나 스크랩하기와 포스트엮기를 기존의 웹 문서 랭킹에 바로 적용할 수 없다. 하이퍼링크는 두 웹 문서 사이의 관계이지만, 블로그 액션은 블로거와 포스트 사이의 관계이기 때문이다. 본 논문에서는 이러한 문제점을 해결하기 위하여 다양한 웹 문서 랭킹 알고리즘들을 블로고스피어에 적합하도록 수정하는 방법을 제안한다. 또한 이 방법을 통해 웹 문서 랭킹 알고리즘들을 수정하여 만들어진 포스트 랭킹 알고리즘들을 소개한다.[1)]

다양한 포스트 랭킹 알고리즘을 소개하고자 하는 이유는 기존에 다양한 포스트 랭킹 알고리즘에 대한 연구가 부족했기 때문이다. 블로그 액션을 이용한 랭킹 알고리즘에 대한 기존의 연구는 블로거 또는 블로그에 랭킹을 부여하는데 주로 초점이 맞추어져 있었다[21][22][23]. 포스트의 품질은 그것을 작성한 블로거의 작성 능력에 따라 결정될 수 있다는 측면에서 블로그의 랭킹이 포스트 랭킹에 확장될 수 있다. 그러나 블로그에는 다양한 주제의 포스트가 저장되어 있고, 그 주제에 따라 블로거의 작성 능력이 다를 수 있기 때문에 블로그에 속한 모든 포스트를 동일하게 평가하는 것은 한계가 있다. 또한, 동일한 주제라 하더라도 블로거는 친분 중시 성향과 정보 중시 성향을 동시에 지니고 있고[18], 친분을 위한 포스트는 검색을 통해 정보를 얻고자하는 사용자의 관점에서는 무의미 할 수 있다. 따라서 동일 블로그의 동일한 주제의 포스트도 각각 그 품질이 다를 수 있다. 이와 같은 이유로 블로그가 아닌 포스트를 위한 랭킹이 포스트 검색에서 중요하다. 그러나 포스트 랭킹 알고리즘에 대한 기존의 연구는 거의 이루어지지 않고 있다.

본 논문에서는 기존의 웹 문서 랭킹 알고리즘 중 유명한 PageRank, HITS 등의 알고리즘을 변형할 뿐만 아니라[6], 기존에 널리 알려지지 않은 HubAVG, ATK, SALSA 등의 웹 문서 랭킹 알고리즘 또한 변형하여 포스트 랭킹 알고리즘으로 소개함으로써 다수의 포스트 랭킹 알고리즘을 소개한다. 널리 알려지지 않은 웹 문서 랭킹 알고리즘 또한 포스트 랭킹에 적용하는 이유는 웹과 블로고스피어의 차이로 인하여 포스트 랭킹에서는 이 알고리즘들이 좋은 성능을 나타낼 가능성이 있기 때문이다. 본 연구에서 제안하는 다양한 알고리즘들은 블로그 액션 중 스크랩하기와 포스트엮기를 이용하기 때문에 스크랩 기능을 지원하는 블로그에 대해서만 적용 가능하다는 한계가 있다. 그러나 스크랩 기능을 지원하는 블로그에서는 본 연구에서 제안하는 알고리즘들을 통하여 정확도 향상을 이룰 수 있다.

각 포스트 랭킹 알고리즘들의 정확도를 평가하고 가장 정확한 알고리즘을 찾기 위하여, 제안하는 알고리즘들과 기존에 소개된 알고리즘인 EigenRumor[5]를 실험을 통해 비교하고자 한다. 이를 위하여 본 논문에서는 스크랩 기능을 지원하는 블로그의 실제 데이터를 수집하고, 이를 통해 랭킹을 부여한 뒤, 사용자들의 의견을 바탕으로 각 랭킹 알고리즘들을 평가하고자 한다. 이러한 연구는 기존의 다양한 웹 문서 랭킹 알고리즘들 중 포스트 랭킹에 가장 적합한 알고리즘을 밝히고, 그들의 특징을 분석함으로써 추후 연구에 도움이 될 수 있다. 또한, 참고문헌 [5]에서와 같이 기존의 연구에서는 포스트 랭킹 알고리즘의 정확도에 대해 수치화하여 나타내는 것이 부족하였으나, 본 논문에서는 각 포스트 랭킹 알고리즘의 정확도를 수치화하여 비교함으로써 차이를 명확하게 비교하고 있다.

본 논문은 다음과 같은 순서로 구성되어 있다. 2장에서는 블로그의 특성에 대해 설명한다. 3장에서는 블로그 액션을 이용한 포스트-블로거 그래프를 정의하고, 기존의 웹 문서 랭킹 알고리즘들을 포스트 랭킹에 적합하도록 수정한 포스트 랭킹 알고리즘을 제안한다. 4장에서는 실제 블로그 데이터를 이용하여 다양한 포스트 랭킹 알고리즘들의 성능을 비교 분석한다. 5장에서는 결론을 내리고, 향후 연구에 대해 논의한다.

1) 이들 중 일부의 기본 아이디어가 참고문헌 [6]에 소개된 바 있다.

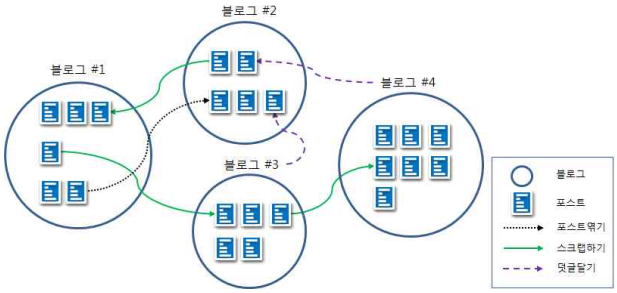
2. 블로그

블로그는 포스트들이 작성시간 순서의 아카이브(archive) 형태로 구성되어 있는 일종의 개인 웹 사이트이다. 블로거는 자신의 블로그에 포스트를 작성하거나 타 블로거가 작성한 포스트들에 다양한 행동을 할 수 있다. 블로거가 자신과 타인의 블로거에 할 수 있는 모든 행동을 본 논문에서는 블로그 액션이라 정의한다. 블로그 액션의 대표적인 예로는 포스트작성, 덧글달기, 스크랩하기, 포스트엮기 등이 있다. 포스트작성은 자신의 블로그에 포스트를 작성하는 기능이며, 타인의 블로그에는 포스트를 작성할 수 없다. 덧글달기는 자신 또는 타인의 포스트의 하단부에 자신의 의견을 짧게 남기는 기능이다. 스크랩하기는 타인의 포스트를 복사하여 자신의 블로그의 포스트로 가져오는 기능이다. 포스트엮기는 자신의 포스트와 타인의 포스트를 엮는 기능으로, 블로거가 자신이 작성하거나 작성중인 포스트의 하단에 동일한 주제를 다루고 있는 타인의 포스트의 주소를 남겨주는 기능이다. 포스트엮기를 통해 서로의 엮인 포스트를 엮인글이라고 한다.

블로고스피어 내의 블로그 액션들을 통해 블로거가 어떤 포스트에 관심을 가졌는지를 알 수 있다. 그러나 덧글달기는 스팸의 목적으로 빈번하게 이용되기 때문에 블로거의 관심과 무관하게 발생할 수 있다는 문제점이 있다. 포스트엮기는 그 기능을 이용하는 과정이 번거롭기 때문에 블로고스피어 내에서 자주 발생하지 않는다. 반면, 스크랩은 다른 블로그 액션들에 비해 블로거가 해당 포스트를 선호한다는 것을 잘 나타낼 뿐만 아니라, 블로고스피어에서 매우 빈번하게 발생하는 액션이다.

(그림 1)은 블로그와 포스트, 블로그 액션으로 구성된 블로고스피어의 예를 나타낸다. 원은 블로그를, 원 안의 사각형은 포스트를 나타낸다. 블로그 액션인 포스트엮기, 스크랩하기, 덧글달기는 가는 점선, 실선, 굵은 점선으로 각각 나타낸다. 블로그 #1, #2, #3, #4의 소유자를 블로거 #1, #2, #3, #4라 하면, (그림 1)의 각 블로그 액션들은 다음과 같은 의미를 지닌다. 블로그 #1의 포스트와 블로그 #2의 포스트 사이의 가는 점선은 블로거 #1이 자신이 작성한 포스트와 블로그 #2의 포스트를 포스트 엮기를 이용하여 엮은 것을 의미한다. 또한, 이 블로그 액션은 블로거 #1이 블로그 #2의 포스트를 엮었다고 할 수 있다. 블로그 #1의 포스트와 블로그 #3의 포스트 사이의 실선은 블로거 #1이 블로그 #3의 포스트를 스크랩하여 그 포스트를 자신의 포스트로 복사하여 가지고 온 것을 의미한다. 블로그 #3와 블로그 #2의 포스트 사이의 굵은 점선은 블로그 #3이 블로그 #2의 포스트에 덧글을 작성한 것을 의미한다.

최근 많은 사람들이 블로그를 이용하여 자신의 의견과 다양한 정보를 포스트로 작성하여 남기고 있다. 이는 다양한 종류의 블로그 액션들이 블로거가 자신의 블로거를 관리하거나 타 블로거들과 서로 의견을 나누기 쉽게 하기 때문이다. 이로 인해 블로그 사용량은 지속적으로 증가하고 있다.



(그림 1) 블로고스피어의 예

따라서 다수의 포스트 중에서 중요한 정보를 담고 있는 포스트를 찾아 높은 랭킹을 부여하는 포스트 랭킹 알고리즘에 대한 연구가 필요하다. 포스트 랭킹 알고리즘에 대한 연구는 블로그 활성화와 정보 검색에 큰 도움이 될 것이다.

3. 관련연구

블로고스피어의 다양한 특징들을 바탕으로 포스트에 랭킹을 부여하는 알고리즘들이 소개되었다. 본 장에서는 기존에 제안된 포스트 랭킹 알고리즘들에 대해 소개하고자 한다.

3.1 EigenRumor

기존에 블로고스피어 환경을 고려한 포스트 랭킹 알고리즘인 EigenRumor가 제안되었다[5]. EigenRumor는 블로거에게 권위점수와 허브점수를 부여하고, 포스트에 평판점수를 부여하였다. 권위점수는 해당 블로거가 작성한 포스트들이 얼마나 높은 평판점수를 가지고 있느냐에 의해 결정된다. 허브점수는 해당 블로거가 덧글을 단 포스트들의 평판점수에 의해 결정된다. 평판점수는 해당 포스트를 작성한 블로거의 권위점수와 덧글을 단 블로거의 허브점수에 의하여 결정된다. 세 가지 점수의 계산은 다음 식으로 정의된다.

$$\vec{r} = \alpha W\vec{a} + (1-\alpha)C\vec{h} \quad \text{(식 1)}$$

$$\vec{a} = W^T\vec{r} \quad \text{(식 2)}$$

$$\vec{h} = C^T\vec{r} \quad \text{(식 3)}$$

벡터 $\vec{r}$, $\vec{a}$, $\vec{h}$는 각각 평판점수, 권위점수, 허브점수를 나타낸다. 행렬 W는 블로거가 포스트를 작성한 관계를 나타내는 행렬이고, 행렬 C는 블로거와 포스트 사이의 덧글을 단 관계를 나타내는 행렬이다. 행렬 W와 C는 다음과 같이 정의된다.

$$W = \begin{cases} w_{i,j} = 1 & \text{블로거 } i\text{가 포스트 } j\text{를 작성한 경우} \\ w_{i,j} = 0 & otherwise \end{cases}$$

$$C = \begin{cases} c_{i,j} = 1 & \text{블로거 } i\text{가 포스트 } j\text{에 덧글을 단 경우} \\ c_{i,j} = 0 & otherwise \end{cases}$$

식 1, 2, 3은 상호참조관계에 있기 때문에 반복하여 계산이 필요하며, 이러한 과정은 HITS[1]에서의 점수 계산과 유사하게 계산된다.

3.2 사용자 피드백과 권위 평가를 이용한 블로그 랭킹 시스템

참고 문헌 [20]에서는 포스트의 적합도와 인기도를 동시에 고려하여 포스트에 랭킹을 부여하는 방법을 제안한다. 적합도는 검색 엔진을 사용하는 사용자가 던진 질의어와 각 포스트가 일치하는 정도를 나타내고, 이는 질의어와 문서 사이의 유사도인 필드별 가중치(Field Weight, FW)와 주제와 문서 사이의 유사도인 분류 가중치(Classification Weight, CW)를 통해 계산된다. 필드별 가중치의 계산에서 매개변수는 휴리스틱(Heuristic)에 의하여 결정되며, 분류 가중치는 베이지안 분류기를 통해 계산된다. 인기도는 포스트의 품질을 나타내는 값을 나타내고, 이는 사용자의 피드백인 북마크 카운트(Bookmark Count, BC)와 클릭 카운트(Click Count, CC)로 계산된다. 북마크 카운트는 블로그를 즐겨찾기한 사용자의 수를 카운트 한 값이고, 클릭 카운트는 포스트를 클릭한 카운트이다.

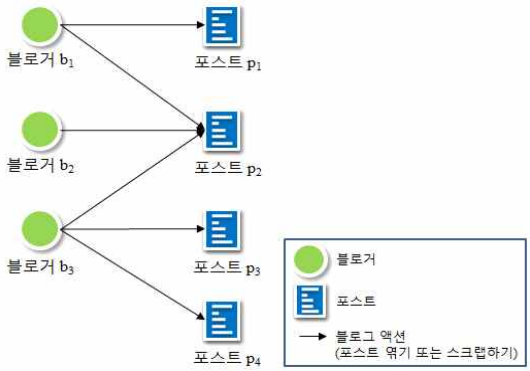
이 논문에서는 FW, CC, BC를 선형모델로 정의하고, 이들의 파라미터(parameter)를 다중 선형 분석을 이용하여 결정한다. 이를 통해 블로거의 점수인 구루 점수(Guru Score, GS)를 계산한다. GS는 FW, CC, BC와 함께 선형모델로 정의되고, 이 모델을 통해 각 포스트의 점수가 계산된다. 이때, 모델의 파라미터는 다중 선형 분석을 이용하여 결정된다.

4. 포스트 랭킹 알고리즘

블로그에는 다양한 블로그 액션들의 존재로 인하여 소수의 하이퍼링크만이 존재한다[5]. 따라서 기존의 웹 문서 랭킹 알고리즘을 포스트 랭킹에 이용할 경우, 대부분의 포스트들의 품질을 정확하게 평가하지 못한다. 이러한 문제를 해결하기 위한 방법으로 본 논문에서는 블로그 액션을 이용하여 포스트에 랭킹을 부여하고자 한다. 블로그 액션은 블로거가 해당 포스트에 대해 만족할 때 발생하기 때문에 하이퍼링크와 동일하게 추천의 의미로 볼 수 있다. 특히 스크랩하기와 포스트엮기는 추천의 의미가 강한 블로그 액션이다. 따라서, 본 논문에서는 스크랩하기와 포스트엮기를 동일하게 추천의 의미로 간주하고, 이를 기반으로 하는 포스트 랭킹 알고리즘을 제안한다.

스크랩하기와 포스트엮기를 통해 포스트에 랭킹을 부여하기 위해서는 블로고스피어를 그래프로 모델링 하여야 한다. 이 그래프에서 블로거는 블로거 노드, 포스트는 포스트 노드로 각각 표현된다. 스크랩하기나 포스트엮기는 (그림 1)과 같이 포스트 사이의 관계로 표현할 수 있으나, 추천의 의미에서 추천을 하는 대상은 포스트가 아닌 블로거이기 때문에 스크랩하기나 포스트엮기는 포스트와 블로거 사이의 관계로 표현되어야 한다. 그러므로 스크랩하기와 포스트엮기는 블로거 노드에서 포스트 노드로 향하는 방향성 에지(directed edge)로 표현된다. 모델링된 그래프를 포스트-블로거 그래프라 부르며, (그림 2)와 같은 이분 그래프(bipartite graph)이다.

(그림 2)는 3명의 블로거와 4개의 포스트로 이루어진 작은 블로고스피어를 포스트-블로거 그래프로 모델링한 예이다. 원은 블로거를, 사각형은 포스트를 각각 나타낸다. 블로거와 포스트 사이의 에지는 스크랩하기 또는 포스트엮기를 나타낸다. 블로거 b_1와 포스트 p_1사이의 에지는 블로거 b_1이 포스트 p_1을 스크랩하거나 엮인글로 엮었다는 것을 의미한다.



(그림 2) 포스트-블로거 그래프

본 논문에서는 기존의 웹 문서 랭킹 알고리즘을 수정하여 이분 그래프의 노드를 평가할 수 있는 포스트 랭킹 알고리즘을 제안한다. 이는 웹 문서 랭킹 알고리즘들은 이분 그래프에 적용될 수 없기 때문이다. 포스트 랭킹 알고리즘은 웹 문서 랭킹 알고리즘과 마찬가지로 각 노드에 점수를 부여되는데, 블로거 노드에 부여되는 점수를 블로거 점수, 포스트 노드에 부여되는 점수를 포스트 점수라 부른다. 포스트 랭킹 알고리즘이 랭킹을 부여하는 과정은 행렬식을 이용하여 표현될 수 있다. 이때, 포스트-블로거 그래프는 다음과 같은 인접행렬 E로 정의된다.

$$E = \begin{cases} e_{i,j} = 1 & \text{블로거 노드 } i\text{와 포스트 노드 } j \text{ 사이에 에지가 존재하는 경우} \\ e_{i,j} = 0 & otherwise \end{cases}$$

인접행렬 E는 m×n 행렬이고, m은 블로거 수를, n은 포스트 수를 각각 나타낸다. 포스트 노드의 점수는 열벡터(column vector) $\vec{p} = [p_1, p_2, \cdots, p_n]^T$, 블로거 노드의 점수는 열벡터 $\vec{b} = [b_1, b_2, \cdots, b_m]^T$로 각각 표현 가능하다. 모든 포스트 랭킹 알고리즘은 인접행렬 E와 포스트 점수 벡터, 블로거 점수 벡터를 이용하여 포스트의 점수를 계산한다.

4.1 PIndegree

PIndegree는 수정 없이 포스트 랭킹에 적용 가능한 알고리즘이다[6]. PIndegree는 많은 블로거들에 의해 여러번 스크랩 당하거나 엮인글로 많이 엮인 포스트를 높은 품질의 포스트로 정의한다. PIndegree에서는 포스트 노드의 차수(degree)를 포스트 점수로 두고, 이에 의하여 랭킹이 결정된다. 포스트 점수의 계산은 다음과 같이 정의된다.

$$\vec{p} = E^T \vec{b_o} \quad \text{(식 4)}$$

식 4에서 $\vec{b_o}$는 블로거 수 m개의 인자로 구성된 열벡터이고, 모든 인자는 1로 초기화 되어 있다.

4.2 PostRank

PostRank는 PageRank의 개념을 블로그 환경에 맞게 변형한 알고리즘이다[6]. PageRank는 높은 품질의 웹 문서가 참조한 웹 문서들이 좋게 평가된다. 이는 높은 품질의 웹 문서를 작성한 작성자의 추천을 통해 높은 품질의 웹 문서를 찾을 수 있을 것이라는 개념을 담고 있는 것이다. PostRank에서는 PageRank의 이러한 개념을 이용하여 좋은 블로거들이 많이 스크랩하거나 엮인글로 엮은 포스트를 품질이 높은 것으로 평가한다. 좋은 블로거란 품질이 높은 포스트를 많이 작성한 블로거를 의미하고, 블로거의 좋은 정도는 블로그 점수로 표현된다.

PostRank에서 블로거 점수를 계산하기 위해서는 블로거와 포스트 사이의 작성 관계가 이용되고, 이는 별도의 작성 그래프로 모델링 될 수 있다. 작성 그래프는 포스트-블로거 그래프와 동일하게 포스트 노드와 블로거 노드로 구성되어 있으며, 에지가 포스트작성인 점만이 다르다. 작성 그래프 또한 행렬로 표현될 수 있으며, 행렬 W는 EigenRumor[5]에서 정의한 행렬과 동일하게 정의된다.

PostRank에서 포스트의 품질은 포스트 점수로 표현된다. 포스트 점수는 포스트 노드와 에지로 연결된 블로거 노드들의 점수의 합에 의해 결정된다. 블로거 점수와 포스트 점수의 계산은 다음 식으로 정의된다.

$$\vec{p} = E^T \vec{b} \quad \text{(식 5)}$$

$$\vec{b} = W\vec{p} \quad \text{(식 6)}$$

식 5와 6에서 벡터 $\vec{p}$와 벡터 $\vec{b}$는 각각 모든 포스트의 점수와 모든 블로거의 점수를 나타낸다. 식 5, 6을 반복하여 HITS와 동일한 방법으로 계산하면 포스트 점수와 블로거 점수가 더 이상 변하지 않는데, 이때의 포스트 점수가 포스트의 품질을 나타내는 점수가 된다. PostRank는 HITS와 달리 블로거 점수와 포스트 점수는 서로 다른 노드에 각각 부여된다는 차이점이 존재한다.

4.3 BAITS (Blog Action Induced Topic Search)

BAITS는 HITS를 블로그 랭킹에 맞도록 변형한 알고리즘이다[6]. BAITS에서는 블로그 액션을 잘 부여하는 블로거들에 의해 추천을 많이 받은 포스트를 품질이 높은 포스트로 정의한다. 블로그 액션을 잘 부여하는 블로거란 좋은 품질의 포스트를 많이 찾아 스크랩하거나 엮인글로 엮은 블로거를 의미하고, 블로거가 액션을 잘 부여하는 정도는 블로거 점수로 나타낸다. 포스트의 품질은 PostRank와 마찬가지로 포스트 점수로 표현된다. 블로거 점수와 포스트 점수의 계산은 다음 두 식에 의해 정의된다.

$$\vec{p} = E^T \vec{b} \quad \text{(식 7)}$$

$$\vec{b} = E\vec{p} \quad \text{(식 8)}$$

식 7과 8를 반복하여 HITS와 동일한 방법으로 계산하여 포스트 점수를 계산하게 된다. BAITS는 HITS와 매우 유사하지만, HITS가 질의어에 해당하는 웹 문서에 대해서만 런타임에 랭킹을 부여한 반면, BAITS는 전체 포스트에 랭킹을 부여한다는 차이점이 있다. 또한 HITS에서는 단일 웹 문서에 권위 점수와 허브 점수가 동시에 부여되었던 반면, BAITS에서는 블로거와 포스트에 각각 한 가지 점수가 부여되었다는 차이점이 있다.

4.4 BAITS에서 블로거 점수 제한

HITS에는 사람이 인위적으로 랭킹을 조작할 수 있다는 단점이 있다. 단순히 많은 웹 문서를 참조하는 것으로 웹 문서는 높은 허브점수를 부여 받을 수 있고, 조작된 높은 허브점수를 이용하여 다른 웹 문서가 높은 권위점수를 가지도록 할 수 있다. 이러한 문제점을 해결하기 위해 HubAVG와 AtK 등이 제안되었다[2]. 두 알고리즘들은 HITS와 동일하게 권위점수를 계산하지만, 허브점수를 다르게 계산한다. HubAVG는 웹 문서 A가 참조하는 모든 웹 문서의 권위점수들의 평균을 웹 문서 A의 허브점수로 부여한다. AtK는 웹 문서 A가 참조하는 모든 웹 문서가 아닌 그 중 가장 권위점수가 높은 k개 웹 문서의 권위점수를 웹 문서 A의 허브점수로 부여한다.

BAITS에서 또한 블로거가 의도적으로 다수의 포스트를 스크랩하거나 엮인글로 엮어서 높은 블로거 점수가 되도록 조작하기 쉽다는 문제점이 있다. 블로거는 의도적으로 조작된 블로거 점수를 통해 특정 포스트를 스크랩하거나 엮인글로 엮음으로써 해당 포스트의 랭킹을 높일 수 있다. 이는 품질이 좋지 않은 포스트에 높은 랭크를 부여하게 만들어 랭킹의 정확도를 감소시키는 요인이 될 수 있다. 본 섹션에서는 이러한 BAITS의 단점을 보완하기 위해 블로거 점수 계산에 제약을 두는 알고리즘들을 소개한다. 이 알고리즘들과 같이 블로거 점수의 계산을 변화시키면 포스트 점수 또한 변화하게 되고, 최종적으로 다른 랭킹 결과를 얻을 수 있다. 이는 블로거 점수를 제한하는 알고리즘들이 HITS와 동일하게 포스트 점수와 블로거 점수를 반복하여 계산하기 때문이다. 따라서 이 알고리즘들은 블로거 점수를 더 정확히 계산함으로써 포스트에 더 정확한 점수가 부여되도록 한다. 제안되는 알고리즘들은 블로거 점수의 계산에만 제약을 두는 방법이기 때문에 포스트 점수의 계산은 BAITS의 방법과 동일하게 계산한다.

4.4.1 BloggerAVG

BloggerAVG는 웹 문서 랭킹 알고리즘 중 HubAVG와

같은 방법을 이용하여 블로거 점수를 제한하는 알고리즘이다. 이 알고리즘은 블로거가 다수의 포스트를 스크랩하거나 엮인글로 엮었어도 그 포스트들이 좋은 품질이 아니라면 블로거 점수가 높게 부여되지 않도록 하는 방안이다. 이를 위해 블로거 점수는 해당 블로거 노드와 에지로 연결된 포스트들의 점수의 평균으로 부여된다. 블로거 점수의 계산은 다음과 같은 식으로 정의된다.

$$\vec{b} = diag(\frac{1}{\sum_{\beta} e_{1,\beta}}, \frac{1}{\sum_{\beta} e_{2,\beta}}, \cdots, \frac{1}{\sum_{\beta} e_{m,\beta}}) E\vec{p} \quad \text{(식 9)}$$

대각선행렬(diagonal matrix)의 각 인자인 $1/\sum_{\beta} e_{i,\beta}$은 행렬 E의 i번째 행의 모든 인자를 더한 값으로, i번째 블로거 노드의 차수를 나타낸다. 식 9를 식 7과 함께 HITS와 같은 방식으로 계산함으로써 BloggerAVG의 결과를 얻을 수 있다.

4.4.2 BloggerAtK

BloggerAtK는 웹 문서 랭킹 알고리즘 중 AtK와 같은 방법으로 블로거의 점수를 제한하는 알고리즘이다. 이 알고리즘 또한 블로거가 다수의 포스트를 스크랩하거나 엮인글로 엮어서 자신의 점수를 높이려는 행위를 막을 수 있다. BloggerAtK는 품질 낮은 포스트를 스크랩하거나 엮인글로 엮더라도 높은 품질의 포스트를 일정 수 이상 스크랩하거나 엮인글로 엮었다면, 블로그 점수를 감점시키지 말아야 한다고 생각하는 방법이다. 이를 위해 블로거가 스크랩하거나 엮인글로 엮은 포스트들 중 가장 높은 포스트 점수를 가진 것들만을 블로거 점수의 계산에 이용한다. 블로거 점수는 다음 식으로 정의된다.

$$b_i = \sum_{j \in F_k(i)} p_j \quad \text{(식 10)}$$

b_i와 p_j는 각각 i번째 블로거 점수와 j번째 포스트 점수를 나타낸다. $F_k(i)$는 i번째 블로거 노드와 연결된 포스트 노드 중 포스트 점수가 높은 k개의 포스트들을 포함하는 집합이다. 식 10과 식 7을 함께 HITS와 같은 방법으로 계산하여 BloggerAtK의 결과를 얻을 수 있다.

4.5 PSALSA

BAITS는 다수의 포스트를 스크랩하거나 엮인글로 엮은 블로거에 높은 점수를 부여하였다. 또한, 다수의 블로거에 의해 스크랩되거나 엮인글로 엮인 포스트에게 높은 점수를 부여하였다. 그러나 다수의 포스트를 스크랩하거나 엮인글로 엮는 것보다, 높은 품질의 포스트만을 선별하여 스크랩하거나 엮인글로 엮는 블로거가 더 좋은 블로거라고 할 수 있다. 또한, 좋은 블로거들이 주로 스크랩하거나 엮인글로 엮는 포스트들이 높은 품질의 포스트라고 할 수 있다. PSALSA는 이러한 개념을 적용한 방법으로, SALSA를 포스트 랭킹에 맞게 적용한 알고리즘이다[4]. PSALSA는 포스트와 블로거 점수 계산에서 두 점수를 모두 제한함으로써 더 정확한 블로거 점수와 포스트 점수를 계산하고자 하는 방법이다. PSALSA의 계산은 다음과 같은 수식으로 정의된다.

$$\vec{b} = E \bullet diag(\frac{1}{\sqrt{\sum_{\beta} e_{\beta,1}}}, \frac{1}{\sqrt{\sum_{\beta} e_{\beta,2}}}, \cdots, \frac{1}{\sqrt{\sum_{\beta} e_{\beta,n}}}) \vec{p} \quad \text{(식 11)}$$

$$\vec{p} = E \bullet diag(\frac{1}{\sqrt{\sum_{\beta} e_{1,\beta}}}, \frac{1}{\sqrt{\sum_{\beta} e_{2,\beta}}}, \cdots, \frac{1}{\sqrt{\sum_{\beta} e_{m,\beta}}}) \vec{b} \quad \text{(식 12)}$$

식 11과 12의 벡터 $\vec{b}$와 $\vec{p}$는 각각 m명의 모든 블로거들의 점수와 n개의 모든 포스트들의 점수를 나타낸다. 식 11과 12를 HITS와 같은 방법으로 계산하여 PSALSA의 결과를 얻을 수 있다. PSALSA와 BloggerAVG는 동일한 방식으로 블로거 점수를 계산하지만, 포스트 점수는 다르게 계산한다. 블로거 점수와 포스트 점수를 HITS와 같은 방법으로 반복하여 계산할 때, 매 반복에서 포스트 점수가 다르게 계산되고, 이는 블로거 점수의 계산 결과를 변화시킨다. 이러한 과정을 통해 최종적으로 PSALSA에서는 BloggerAVG와 다른 결과를 얻게 된다.

5. 실험

본 장에서는 소개된 포스트 랭킹 알고리즘들의 성능을 실험을 통해 평가하고, 성능 평가 결과를 통하여 정확한 포스트 랭킹을 부여할 수 있는 알고리즘을 선별한다.

5.1 실험 환경

실험을 위해 국내 블로그 서비스 중 하나인 네이버 블로그에서 2006년 4월부터 수개월간 수집하여 익명으로 처리한 데이터를 사용하였다. 본 실험에서는 질의와 포스트의 연관성을 제외하고, 포스트 랭킹 알고리즘이 포스트의 품질을 잘 평가 하였는지를 확인하고자 한다. 이를 위해 20개의 질의어를 바탕으로 질의어와 연관성이 높은 포스트를 수집하여, 질의어 별로 20개의 포스트 집합을 생성하였다. 사용된 질의어는 A. Borodin[2]와 J. Kleinberg[1]가 사용한 질의를 바탕으로 한 <표 1>의 20개 질의를 사용하였다. 각 질의별 포스트 집합을 바탕으로 포스트 랭킹 알고리즘에서 추천하는 포스트들과 그 포스트에 대한 사용자의 평가를 통해 포스트 랭킹 알고리즘의 정확도를 보이고자 한다.

〈표 1〉 질의어

알콜 중독자, 뚝섬 유원지, 루이 암스트롱, 닛산 자동차, 길거리 농구, 클래식 기타, 사형제도 반대, 걸프전, 유전자 조작, 마이클 조던, 달 착륙, 이터널 선샤인, 설악산 국립공원, 인터넷 검열, 찬밥 요리, 검색 엔진 Google, 셰익스피어, 우표 수집, 병렬 구조, 타이 관광

포스트 랭킹 알고리즘의 성능을 측정하기 위해서는 사용자의 의견을 반영하여 포스트의 실제 품질을 평가해야 한다. 본 실험에서는 11명의 평가자에게 포스트의 품질을 '최상', '상', '하'로 평가하도록 하였다. 검색 결과의 상단에 반드시 나타나야 한다고 판단되는 포스트는 '최상'으로 평가하도록 하였다. 또한 검색 결과의 상단이나 그 근처에 나타나야 한다고 판단되는 포스트는 '상'으로 평가하도록 하였다. 마지막으로 검색 결과의 상단에서 떨어진 위치에 나타나야 된다고 판단되는 포스트는 '하'로 평가하도록 하였다. 따라서 '최상'과 '상'으로 평가된 포스트는 포스트 랭킹 알고리즘에서 추천되어야 하는 포스트들이라 할 수 있다. 포스트 랭킹 알고리즘에 의하여 상위 랭크로 선택한 포스트들이 사용자에 의해 '최상' 또는 '상'으로 평가되었다면, 해당 알고리즘의 정확도는 높은 것으로 판단할 수 있다. 이와 같은 평가자를 통한 실험은 기존의 웹 문서 랭킹 알고리즘에 관한 연구[1][14][16][19]에서 주로 이용되었던 방법이다.

포스트 랭킹 알고리즘의 정확도는 정밀도(precision)[7]와 평균 정밀도(average precision)[8]를 통해 나타냈다. 정밀도와 평균 정밀도는 각 20개의 질의어 별로 계산되고, 20개의 정밀도와 20개의 평균 정밀도의 평균을 각 포스트 랭킹 알고리즘의 정확도로 간주하였다. 정밀도와 평균 정밀도의 각 평균을 정확도로 간주하는 이유는 각 질의별 정밀도와 평균 정밀도의 차이가 크지 않기 때문이다.

정밀도와 평균 정밀도를 측정하기 위해서는 평가자의 평가를 바탕으로 정답 포스트를 선정해야 한다. 평가자가 '최상' 또는 '상'으로 평가한 포스트가 품질이 좋은 포스트이기 때문에 본 논문에서는 정답 포스트를 평가자들 중 다수가 '최상'으로 평가하거나 '최상' 또는 '상'으로 평가한 포스트로 정의한다. '최상'인 평가만으로 정답 포스트를 선정하는 이유는 알고리즘의 정확도를 평가할 때, 더 엄격한 평가를 하기 위함이다. 본 실험에서 정밀도와 평균 정밀도는 각 질의별로 각 포스트 랭킹 알고리즘들이 선정한 상위 5위 또는 10위까지의 포스트들과 평가자의 평가 결과를 비교하여 측정하였다.

정밀도는 각 포스트 랭킹 알고리즘이 가장 높게 평가한 상위 5개 또는 10개의 포스트들 중 정답 포스트가 얼마나 되는지 그 비중을 측정한 값이다. 이는 다음과 같은 식으로 정의된다.

$$\text{precision} = \frac{|\text{high quality posts} \cap \text{recommended posts}|}{|\text{recommended posts}|}$$

(식 13)

식 13에서 {recommended posts}는 포스트 랭킹 알고리즘이 높은 점수를 부여하여 상위 5개 또는 10개 안에 포함된 포스트 집합이다. {high quality posts}는 정답 포스트들의 집합을 의미한다. 정밀도는 상위 n개의 결과에 대해 평가를 할 때, p@n으로 기술하기도 한다. 즉, 본 논문에서는 p@5와 p@10의 결과를 보이고자 한다.

평균 정밀도는 정밀도와 유사한 개념이지만, 상위 랭크를 잘 평가할수록 더 높은 값으로 평가하는 척도이다. 평균 정밀도는 포스트 랭킹 알고리즘이 포스트들에 부여한 랭크를 바탕으로 1위부터 순서대로 확인하여 '최상' 또는 '상'으로 평가된 포스트의 랭크에서의 정밀도를 계산하고, 이렇게 계산된 모든 정밀도들의 평균을 결과로 보이는 방법이다. 상위 n개의 랭킹 결과의 평균 정밀도는 다음과 같은 식으로 정의할 수 있다.

$$\text{average precision} = \frac{\sum_{i=1}^{n} (p@i \times \text{HighQualty}(i))}{|\text{high quality posts} \cap \text{recommended posts}|}$$

(식 14)

식 14에서 HighQuality(i) 함수는 포스트 랭킹 알고리즘에서 i위로 부여된 포스트가 정답 포스트인지를 알려주는 함수이다. i위 포스트가 정답 포스트이면 1, 그렇지 않다면 0을 반환하게 된다. 평균 정밀도 또는 정밀도와 마찬가지로 상위 5, 10개의 포스트를 대상으로 측정하며, 그 결과가 높을수록 알고리즘의 정확도가 높다고 할 수 있다.

본 실험에서는 EigenRumor[5], PIndegree[6], PostRank[6], BAITS[6], BloggerAVG, BloggerAtK, PSALSA의 정밀도와 평균 정밀도를 측정하였다. PIndegree를 제외한 나머지 랭킹 알고리즘은 포스트 점수와 블로거 점수가 HITS에서처럼 반복하여 계산되는 방법들로, 포스트 점수의 변화가 10-8보다 작아질 때까지 반복하여 계산하였다. BloggerAtK의 파라미터 K는 블로거 노드들의 차수의 평균과 차수의 중앙값(median)을 각각 이용하였다. 평균과 중앙값을 이용하는 방법의 결과는 각각 BloggerAtK(AVG), BloggerAtK(MED)로 표시 하였다. 참고문헌 [20]에서 제안한 방법은 본 논문의 실험에서 제외하였는데, 이는 본 논문에서는 포스트의 품질에 대해서만 관심이 있지만, 참고문헌 [20]에서는 포스트의 품질과 함께 포스트와 키워드의 적합성을 동시에 고려하는 랭킹 알고리즘을 제안하고 있기 때문이다.

5.2 실험 결과

<표 2와 3>은 각 포스트 랭킹 알고리즘의 정밀도[7]와 평균 정밀도[8]의 평균을 나타낸 결과이다. 각 표에는 알고리즘별로 4개의 결과가 나타나 있다. 첫 번째 열과 두 번째 열은 '최상'으로 평가된 포스트를 정답 포스트로 간주할 때, 각 알고리즘이 제시한 상위 5개와 10개 포스트의 결과에 대한 정밀도와 평균 정밀도의 평균을 의미한다. 세 번째 열과 마지막 열은 '최상' 또는 '상'으로 평가된 포스트를 정답 포스트로 간주할 때, 각 알고리즘이 제시한 상위 5개와 10개 포스트의 결과에 대한 정밀도와 평균 정밀도의 평균을 의미한다.

<표 2>는 각 알고리즘의 정밀도를 나타낸 결과이다. '최상'으로 평가된 포스트를 정답 포스트로 간주한 경우의 정밀도에서는 BloggerAVG가 종합적으로 가장 정확한 것으로 평가되었다. 또한 BAITS와 EigenRumor도 그 다음으로 정확한 것으로 평가되었다. 반면, PostRank와 BloggerAtK

(MED)는 가장 낮은 정밀도를 보였다. '최상'과 '상'으로 평가된 포스트를 정답 포스트로 간주한 경우에서는 BloggerAVG가 높은 정밀도를 보였다. 또한 BAITS, EigenRumor도 그 다음으로 정확한 것으로 평가되었다. 반면, BloggerAtK(MED), PIndegree는 낮은 정밀도를 보였다.

<표 2>에서 BAITS는 높은 정밀도를 보였다. 이는 품질이 높은 포스트를 잘 찾아 스크랩하거나 엮인글로 엮는 블로거의 평가를 높게 생각하겠다는 개념이 포스트 랭킹에 큰 도움이 되기 때문이다. 또한, BAITS에서 블로거 점수를 제한하는 방법인 BloggerAVG는 BAITS보다 더 높은 정밀도를 보였다. 이는 BloggerAVG가 포스트의 품질에 관계없이 다수의 포스트를 스크랩하거나 엮인글로 엮은 블로거를 능력이 부족한 블로거로 보고, 포스트 평가시에 이들의 영향력을 감소시키기 때문이다. 그 결과로 능력이 부족한 블로거의 영향으로 인해 상위에 랭크되었던 포스트가 제거됨으로써, 품질이 좋은 포스트가 더 상위에 랭크 될 수 있게 된다.

그러나 BAITS에서 블로거 점수를 제한하는 또 다른 알고리즘인 BloggerAtK는 높은 정밀도를 보이지 못하였고, 특히 BloggerAtK(MED)는 대부분 경우에 가장 낮은 정밀도를 보였다. 이는 BloggerAtK가 블로거가 스크랩하거나 엮인글로 엮은 스크랩 중 품질이 가장 좋은 포스트 일부만을 이용하여 그 블로거의 능력을 평가하기 때문이다. 따라서 블로거가 스크랩하거나 엮인글로 엮은 포스트 중에서 품질 낮은 포스트가 다수 존재한다면 그 블로거의 능력을 낮게 평가해야만 더 나은 포스트 랭킹 결과를 얻을 수 있다. 또한, BloogerAtK(AVG)와 BloggerAtK(MED)의 차이를 통해 k값에 따라 알고리즘의 정밀도가 차이가 있음을 알 수 있다. 이는 k값을 적절하게 부여함에 따라 BloggerAtK가 더 나은 결과를 보일 수 있다고 해석할 수 있다.

<표 2>에서 PIndegree는 다른 방법들보다 낮은 정밀도를 보였다. 이는 자신의 노드의 차수뿐만 아니라 자신의 주변에 있는 노드의 차수 또한 고려하여 랭킹을 계산하는 것이 포스트 랭킹에서 더 정확한 결과를 보일 수 있기 때문이다. PostRank도 낮은 정밀도를 보였는데, 이는 포스트를 잘 작성하는 블로거들이 평가 또한 잘 할 것이라는 개념이 포스트 랭킹에 적합하지 않기 때문으로 판단된다.

종합적으로 BloggerAVG, BAITS가 높은 정밀도를 보였고, 특히 BloggerAVG는 대부분 경우에 높은 정밀도를 보였다. 반면, BloggerAtK(MED), PostRank, PIndegree는 낮은 정확도를 보였다. 특히, BloggerAtK(MED)는 대부분 경우에 낮은 정확도를 보였다.

<표 3>은 각 알고리즘의 평균 정밀도를 나타낸 결과이다. '최상'으로 평가된 포스트를 정답 포스트로 간주한 경우, PIndegree, BloggerAVG, PSALSA 등이 좋은 결과를 보였다. 반면, PostRank와 BloggerAtK(MED)는 정밀도에서의 결과와 동일하게 낮은 평균 정밀도를 보였다. '최상'과 '상'으로 평가된 포스트를 정답 포스트로 간주한 경우, 상위 5개와 10개의 포스트에 대한 평균 정밀도에서는 BloggerAVG가 높게 평가되었다. 반면, PostRank와 BloggerAtK(MED)

〈표 2〉 포스트 랭킹 알고리즘들의 정밀도

	High		High or Good	
	at 5	at 10	at 5	at 10
PIndegree	0.38	0.395	0.82	0.83
PostRank	0.36	0.395	0.82	0.855
BAITS	0.46	**0.45**	0.88	**0.865**
BloggerAVG	**0.5**	0.4	**0.93**	**0.905**
BloggerAtK(AVG)	0.43	0.425	0.85	0.825
BloggerAtK(MED)	0.37	0.35	0.79	0.815
PSALSA	0.38	0.41	0.82	0.835
EigenRumor	0.44	**0.445**	0.88	**0.865**

〈표 3〉 포스트 랭킹 알고리즘들의 평균 정밀도

	High		High or Good	
	at 5	at 10	at 5	at 10
PIndegree	**0.699**	0.582	0.900	0.871
PostRank	0.418	0.457	0.817	0.839
BAITS	0.627	0.577	0.892	0.881
BloggerAVG	0.650	**0.622**	**0.978**	**0.953**
BloggerAtK(AVG)	0.652	0.554	0.879	0.868
BloggerAtK(MED)	0.510	0.498	0.862	0.846
PSALSA	**0.695**	0.580	0.900	0.870
EigenRumor	0.628	0.576	0.905	0.898

는 낮은 평균 정밀도를 보였다. 평균 정밀도에서는 종합적으로 BloggerAVG가 가장 좋은 것으로 평가되었고, PostRank와 BloggerAtK(MED)가 좋지 않게 평가되었다.

PIndegree는 가장 낮은 정밀도를 보였지만 평균 정밀도에서는 좋은 결과를 보였고, PSALSA도 정밀도는 높지 않았지만 높은 평균 정밀도를 보였다. 이는 PIndegree와 PSALSA가 1위 또는 2위에 '최상'으로 평가된 포스트를 많이 랭크 하였고, 그 외의 순위에는 '최상'으로 평가된 포스트를 거의 랭크 하지 못하였기 때문이다. 반면, BAITS는 정밀도에서 높은 성능을 보였지만 평균 정밀도에서는 높은 정확도를 보이지 못하였다. 이는 BAITS가 1위 또는 2위에 '최상' 또는 '상'으로 평가된 포스트를 많이 추천하지 못하였기 때문이다. BloggerAVG는 정밀도에서와 유사하게 평균 정밀도에서도 좋은 결과를 보였다. 이를 통해 BloggerAVG가 상위 5, 10위 내에 품질이 좋은 포스트를 잘 추천할 뿐만 아니라, 특히 1위 또는 2위의 랭크에 품질이 좋은 포스트들을 잘 추천한다는 것을 알 수 있다. PostRank와 BloggerAtK(MED)는 정밀도에서와 동일하게 낮은 평균 정밀도를 보였다. 이를 통해 두 알고리즘이 1위 또는 2위에도 품질 좋은 포스트를 추천하지 못한다는 것을 알 수 있다.

정밀도와 평균 정밀도의 결과를 종합적으로 봤을 때, 본 논문에서 추천하는 알고리즘은 BloggerAVG이다. 이 방안은 대부분의 경우에서 높은 정확도를 보였다. 특히, BloggerAVG는 '최상'으로 평가된 포스트만을 정답으로 간주한 경우에서도 대부분 높은 정확도를 보였는데, 이는 BloggerAVG가 포스트 랭킹에서 매우 높은 정확도를 보인

다는 것을 의미한다. 또한, BloggerAVG가 BAITS보다 높은 정확도를 보였는데, 이는 BAITS를 그냥 이용하는 것보다는 포스트 랭킹에 맞도록 수정하는 것이 더 나은 성능을 보일 수 있다는 것으로 추측된다.

반면, PostRank와 BloggerAtK(MED)는 대부분의 경우에서 낮은 정확도를 보였다. PostRank가 낮은 정확도를 보이는 것은 포스트를 잘 작성하는 블로거들이 평가 또한 잘 할 것이라는 개념 때문이다. PostRank의 이러한 개념은 EigenRumor에도 일부 이용되었지만, EigenRumor는 높은 정밀도를 보였다. 이는 EigenRumor가 포스트작성 뿐만 아니라 덧글달기를 이용하였다는 차이로 인해 발생하는 것으로 보인다. 즉, PostRank와 EigenRumor의 결과로 미루어 볼 때, 양질의 포스트를 작성하는 블로거는 양질의 포스트를 스크랩하거나 엮인글로 엮는 일 보단 덧글을 다는데 더 관심이 있을 것이라는 추측 할 수 있다. 이는 포스트작성과 덧글달기가 모두 글을 작성하는 행위라는 점에서 유사하기 때문에 타당성이 있을 것으로 판단된다. BloggerAtK(MED) 또한 낮은 정확도를 보이는데, BloggerAtK(AVG)의 정확도는 낮지 않은 것으로 볼 때, 블로거 점수를 일부만 이용할 때 몇 개의 블로거를 이용할 것인지에 대한 파라미터 K의 설정이 중요함을 알 수 있다.

6. 결론 및 향후 연구

최근 웹에서는 블로그 사용이 활발히 이루어지고 있고, 이로 인해 다수의 포스트가 생성되고 있다. 이러한 상황에서 포스트 및 블로그 검색에 대한 연구들이 진행되고 있었으며, 특히 포스트에 랭킹을 부여하는 포스트 랭킹 알고리즘에 대한 연구가 필요한 상황이다. 포스트는 일종의 웹 문서라고 볼 수 있기 때문에 웹 문서 랭킹 알고리즘들을 통해 랭킹을 부여할 수 있다. 기존의 웹 문서 랭킹 알고리즘들은 하이퍼링크를 이용하여 랭킹을 부여하지만 포스트 사이에는 하이퍼링크가 적게 존재하므로 웹 문서 랭킹 알고리즘들은 소수의 포스트에만 랭킹을 부여할 수 있다. 따라서 포스트 랭킹에 적합한 포스트 랭킹 알고리즘에 대한 연구가 필요하다. 그러나 기존에 다양한 포스트 랭킹 알고리즘이 제안되지 못 하였다.

본 논문에서는 다양한 포스트 랭킹 알고리즘을 제안하기 위하여 다양한 블로그 액션들 중 스크랩하기와 포스트엮기를 통해 포스트에 랭킹을 부여하고자 하였다. 이를 위하여 블로고스피어를 블로거, 포스트, 스크랩과 엮인글 관계로 이루어진 포스트-블로거 그래프로 모델링 하였다. 또한, 기존의 웹 문서 랭킹 알고리즘들은 포스트-블로거 그래프에 맞지 않기 때문에 수정이 필요하다. 본 논문에서는 기존의 웹 문서 랭킹 알고리즘들을 포스트-블로거 그래프에 맞도록 수정한 다양한 포스트 랭킹 알고리즘들을 제안하였다. 제안한 포스트 랭킹 알고리즘들과 기존의 포스트 랭킹 알고리즘을 스크랩 기능을 지원하는 블로그의 실제 데이터와 유저 스터디를 통하여 정확도를 측정하였다. 그 결과 BloggerAVG의 정확도가 높은 것으로 평가되었고, PostRank와 BloggerAtK(MED)의 정확도가 낮은 것으로 평가되었다.

본 논문에서는 블로그 액션 중 추천의 의미를 잘 나타내는 것으로 생각되는 스크랩과 엮인글 생성을 이용하는 포스트 랭킹 알고리즘을 제안하였다. 그러나 블로고스피어에는 스크랩과 엮인글 생성 이외에 다양한 블로그 액션들이 존재하고, 이것은 포스트 랭킹에 도움 될 수 있다. 특히 덧글달기와 포스트작성을 이용한 EigenRumor의 정확도가 높은 것으로 측정되었기 때문에 제안하는 포스트 랭킹 알고리즘에 덧글쓰기 등의 다양한 블로그 액션을 접목하여 정확도를 향상시키고자 하는 방법을 제시할 수 있다. 또한 블로고스피어 내에는 포스트가 생성된 시각, 블로그 액션이 생성된 시각 정보가 기록되어 있다. 이는 웹 문서와 다른 블로고스피어의 특징으로써 이런 정보를 활용하여 포스트 랭킹 알고리즘의 정확도를 향상 시키는 방법을 제시할 수 있다. 또한 BloggerAtK는 파라미터 K에 따라 그 정확도의 차이가 뚜렷한데, 향후 연구로 최적의 파라미터 K에 대해 알아보고자 한다.

참 고 문 헌

[1] J. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” In *Proc. of the 9th ACM-SIAM Symp. on Discrete Algorithms*, pp.668 - 677, 1998.

[2] A. Borodin, R. Gareth, S. Jeffrey, and T. Panayiotis, “Link Analysis Ranking: Algorithms, Theory, and Experiments,” *ACM Transactions on Internet Technology*, 5(1):231 - 297, 2005.

[3] S. Brin and L. Page, “The Anatomy of a Large-scale Hypertextual Web Search Engine,” In *Proc. of the 7th Int'l Conf. on World Wide Web*, WWW, pp.107 - 117, 1998.

[4] R. Lempel and S. Morgan, “The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect,” In *Proc. of the 9th Int'l Conf. on World Wide Web*, WWW, pp.387 - 401, 2000.

[5] K. Fujimura, T. Inoue, and M. Sugisaki, “The Eigenrumor Algorithm for Ranking Blogs,” In *Proc. of the 14th Int'l Conf. on World Wide Web*, WWW, 2005.

[6] W. Hwang, S. Kim, D. Bae and Y. Do, “Post Ranking Algorithms in Blog Environment,” In *Proc. of the 2nd Int'l Conf. on Future Generation Communication and Networking Symposia*, pp.64-67, 2008.

[7] V. Rijsbergen, C.J. *Information Retrieval*. 2nd edition, 1979, London, Butterworths.

[8] S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*, Morgan-Kaufmann Publishers, 2002.

[9] T. Haveliwala, “Topic Sensitive Pagerank,” In *Proc. of the 11th Int'l Conf. on World Wide Web*, WWW, pp.517-526, 2002.

[10] A. Ng, A. Zheng, and M. Jordan, “Link Analysis, Eigenvectors, and Stability,” In *Proc. of the 17th Int'l Joint Conf. on Artificial Intelligence*, pp.903 - 910, 2001.

[11] L. Nie, B. Wu, and B. D. Davison, “A Cautious Surfer for Pagerank,” In *Proc. of the 16th Int'l Conf. on World Wide Web*, WWW, pp.1119 - 1120, 2007.
[12] D. Rafiei and A. Mendelzon, “What is This Page Known for? Computing Web Page Reputations,” In *Proc. of the 9th Int'l Conf. on World Wide Web*, WWW, pp.823 - 835, 2000.
[13] M. Richardson and P. Domingos, “The Intelligent Surfer: Probabilistic Combination of Link and Content Information in Pagerank,” In *Advances in Neural Information Processing Systems*, pp.1441 - 1448, 2002.
[14] K. Bharat and M. Henzinger, “Improved Algorithms for Topic Distillation in a Hyperlinked Environment,” In *Proc. of the 21st annual int'l ACM SIGIR conf. on Research and development in information retrieval*, pp.104 - 111, 1998.
[15] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan, “Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text,” In *Proc. of the 7th Int'l Conf. on World Wide Web*, WWW, pp.65 - 74, 1998.
[16] M. Diligenti, M. Gori, and M. Maggini, “Web Page Scoring Systems for Horizontal and Vertical Search,” In *Proc. of the 11th Int'l Conf. on World Wide Web*, WWW, pp.508-516, 2002.
[17] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon, “PageRank, HITS and a Unifed Framework for Link Analysis,” In *Proc. of the 25th ACM SIGIR Conf.*, pp.353-354, Tampere, Finland, Aughust, 2002.
[18] S. Yoon, S. Kim, and S. Park, "Determining the Strength of the Propensities of a Blog Network," In *Proc. IEEE International Symp. on Computational Intelligence and Data Mining*, pp.140-145, Sheraton Music City Hotel, Nashville, TN, USA, Mar. 30 - Apr. 2, 2009.
[19] P. Lynch, X. Luan, M. Prettyman, L. Mericle, E. Borkmann, and J. Schlaifer, "An Evaluation of New and Old Similarity Ranking Algorithms," In *Proc. Int'l Conf. on Information Technology: Coding and Computing*, pp.148-149, 2004.
[20] K. Jeong, "Blog Rank System Using User Feedback and Authority Estimation," Ph. D. Dissertation, Graduate School Chonnam National University, 2009.
[21] E. Adar, L. Zhang, L. Adamic, and R. Lukose, “Implicit Structure and the Dynamics of Blogspace,” *Workshop on the Weblogging Ecosystem at the 13th Int'l Conf. on World Wide Web*, WWW, 2004.
[22] K. Apostolos, S. Martha, and V. Iraklis, “BlogRank: Ranking Weblogs based on Connectivity and Similarity Features,” In *Proc. of the 2nd int'l workshop on Advanced architectures and algorithms for internet delivery and applications*, AAA-IDEA, 2006.
[23] A. T. Mohamad, S. M. Hashemi, and M. Ali, “B2Rank: An Algorithm for Ranking Blogs Based on Behavioral Features,” In *Proc. of the IEEE/WIC/ACM Int'l Conf. on Web Intelligence*, pp.104-107, 2007.

황 원 석

e-mail : hws23@agape.hanyang.ac.kr
2007년 한양대학교 정보통신학부 컴퓨터 전공(학사)
2009년 한양대학교 정보통신대학원 (공학석사)
2009년～현 재 한양대학교 전자통신 컴퓨터공학과 박사과정 재학중
관심분야 : 사회연결망분석, 인터넷 포탈 데이타 분석, e-비지니스, 데이타 마이닝

도 영 주

e-mail : trinity@agape.hanyang.ac.kr
2005년 한양대학교 전자전기컴퓨터공학부 (학사)
2008년 한양대학교 정보통신대학원 (공학석사)
2008년～현 재 매크로임팩트(주) 연구원
관심분야 : 사회연결망분석, 인터넷 포탈 데이타 분석, e-비지니스, 데이타 마이닝

김 상 욱

e-mail : wook@hanyang.ac.kr
1989년 2월 서울대학교 컴퓨터공학과 (학사)
1991년 2월 한국과학기술원 전산학과 (석사)
1994년 2월 한국과학기술원 전산학사 (박사)
1991년 7월～1991년 8월 미국 Stanford University, Computer Science Department, 방문 연구원
1994년 3월～1995년 2월 KAIST 정보전자연구소 전문 연구원
1999년 8월～2000년 8월 미국 IBM T.J. Watson Research Center, Post-Doc.
1995년 3월～2003년 2월 강원대학교 정보통신공학과 부교수
2009년 1월～2010년 2월 미국 Carnegie Mellon University, Visiting Scholar
2003년 3월～현 재 한양대학교 정보통신대학 정보통신학부 교수
관심분야 : 데이터베이스 시스템, 저장 시스템, 트랜잭션 관리, 데이터 마이닝, 멀티미디어 정보 검색, 공간 데이터베이스/GIS, 주기억장치 데이터베이스, 이동 객체 데이터베이스/텔레매틱스, 사회 연결망 분석, 웹 데이터 분석