지능정보연구 제15권 제2호 2009년 6월(pp.61~74)

주제 유사성 기반 클러스터링을 이용한 블로그 검색기법 연구

이기준
연세대학교 공과대학 정보산업공학과
(backdale@naver.com)

이명진
연세대학교 공과대학 정보산업공학과
(xml@yonsei.ac.kr)

김우주
연세대학교 공과대학 정보산업공학과
(wkim@yonsei.ac.kr)

웹 2.0에 기반한 정보화 사회에 있어 참여를 통한 자료의 축적 속도는 더욱 더 가속화 되어가고 있다. 이러한 현상 속에서, 웹 2.0으로 인해 정보의 저장 및 공유 형태 역시 단순 웹 페이지에서 블로그로 나아가 포드캐스팅, 비디오 등의 다양한 모습으로 분화되어가고 있는 실정인데, 이는 웹 상의 정보에 대한 통합적이고 효율적인 접근을 오히려 방해할 수 있는 요소이기에 보다 효과적인 정보 검색 방법을 요구하게 된다.

본 연구에서는 특히 블로그 검색에 초점을 맞추어 기존 웹 검색 방식의 문제점을 도출, 해결하고자 한다.

논문에서 제안하고자 바는 특정 검색어에 대해 블로그 검색을 수행한 후, 검색 결과에서 주요 주제들을 효과적으로 추출하고, 주제별로 결과물들을 클러스터링하여 순위별로 제공하고자 하는 것이다. 이를 통해 블로그 검색에의 정보 추출에서 사용자에게 특정 검색어에 대해 보다 동적인 추가 주제 카탈로그를 제시함으로써 대량의 의미 없는 정보들을 단순 브라이징하는 방식을 벗어날 수 있으며, 빠르게 검색 의도에 유의한 자료들에 접근할 수 있도록 할 수 있다.

논문접수일 : 2008년 04월 02일 논문수정일 : 2009년 03월 03일 게재확정일 : 2009년 06월 13일 교신저자 : 김우주

1. 서론

인터넷의 탄생과 함께 웹에 수많은 정보를 저장할 수 있게 되면서 인터넷을 통해 얻는 정보들은 사회적으로 중요한 비중을 차지하게 되었다. 이런 배경의 넘치는 정보들 속에서 좋은 정보와 안 좋은 정보를 구별하는 것이 중요해 졌으며 이를 위한 검색엔진이 탄생하게 되었다. 더욱이 웹 2.0에 기반한 현재의 웹에서는 사용자의 참여를 통한 정보의 축척이 늘어나면서(Tim O'reilly, 2004), 정보의 양이 훨씬 빠른 속도로 늘어나게 되었고, 이러한 대량의 정보를 효율적으로 활용할 수 있는 검색 방법 역시 함께 요구되고 있는 실정이다. 이에 따라, 본 연구에서는 특히 블로그 검색에 초점을 맞추어 기존 웹 검색 방식의 문제점을 도출하고 그것을 해결하고자 한다.

블로그(Blog)라는 용어는 웹(Web, 웹)과 로그(Log, 일지)의 합성어로 1999년 Cameron Barett이 쓴 에세이 "Anatomy of Weblog"에서 처음으로 사용되었다(김재한, 옥철영, 1994). 최근 사람들이 인터넷을 이용하는 것이 일반화되었고 웹 상에서 개개인의 비중이 커지게 되었다. 따라서 자신이 자주 이용하는 웹이라는 공간에 자신의 관심사나 감정, 개성을 표출하고 나누고 싶어하는 성향 또한 매우 일반화 됨에 따라(권오병, 문예성, 김민용, 2007) 블로그의 형태도 다양해지고 있으며, 새로운 정보

를 손쉽고 빠르게, 자유롭게 올릴 수 있다는 점, 이용자의 수가 많다는 점 등 많은 이점들을 가지고 있어 잠재력도 무궁무진하다고 할 수 있겠다.

이런 블로그들은 개인적인 목적으로 생성이 되고 개인적인 정보들이 올라오기 때문에(최호찬, 2004) 일반적인 웹 사이트들이 가지고 있는 정보들과는 다르다. 따라서 이런 정보들을 보다 효율적으로 검색하여 제시할 수 있다면 기존의 웹이 줄 수 있는 것과는 또 다른 성질의 유용한 정보를 제공하게 됨으로 본 연구가 의의를 가질 수 있게 된다.

그러나 기존의 검색 방식으로 블로그들을 검색하여 보면, 사용자가 원하는 정보를 얻기가 어렵다. 예를 들어 사용자가 '에쿠니 가오리' 라는 소설가에 대하여 처음 알게 되었고, 이에 따라 어떤 책을 먼저 읽을지, 대중들은 그녀에 대해 어떤 인식을 하고 있고, 어떤 서평을 남기고 있는지 궁금하여 검색을 해보았다고 가정하자. 그러나 현재의 검색엔진이 제공하는 검색결과를 통해서는 위와 같은 검색 동기를 만족하기까지는 상당한 시간이 소요된다. 무엇보다도 자료들에 아무런 분류나 비중을 두지 않고, 그저 텍스트가 일치하는 선에서 구분없이 제시되고 있기 때문이다. 그 중에는 실제로는 작가와 관련이 없는 자료들도 많이 나열 될 수 있고, 책과 상관이 없는 다른 분야의 자료들도 상당 수 포함 되어 있다. 따라서 사용자들은 자신이 원하는 정도의 정보를 얻기 위해서는 상당한 시간을 들여서 여러 페이지를 살펴보아야만 정보의 분포 정도를 파악하게 된다. 이러한 현상은 동명이인이 있는 경우나, 여러가지 분야에서 활동하고 있는 인물에 대해 검색하였을 때에는 그 정도가 더욱 심해진다. 예를 들어, '유이' 라는 가수에 대해 알고 싶어서 검색어를 입력하고 결과를 받아보면, 조금 더 인지도가 있는 배우 '아라가키 유이' 때문에 꽤나 많은 페이지를 넘겨도 충분한 자료를 모으기 어렵다. 실제로 2008-04-22 현재-유이란 이름이 흔한 탓도 있지만-1페이지의 하나 꼴로만 원하는 정보를 볼 수 있었다.

일반적으로 어떠한 검색어를 입력하더라도 위와 같은 문제들은 빈번하게 일어난다. 왜냐하면, 현재 모든 일반적인 키워드들은 여러 분야에 걸쳐 의미를 가지고 있으며, 동음의 다른 뜻을 가진 단어들도 존재하기 때문이다. 특히 가장 커다란 자율성을 가진 블로그 정보들이 아무런 비중을 두지 않고, 분류되지 않은 상태로 제공 된다는 것은 사용자가 손쉽게 좋은 정보를 얻는데 있어 장애가 되는 요소가 아닐 수 없다.

본 논문에서는 이러한 문제를 해결하기 위해 주제 유사성을 기반으로 한 클러스터링 기법을 통해 블로그 자료를 분류하는 검색방법을 제안하고자 한다. 제 2장에서는 논문의 기반이 되는 관련연구에 대하여 살펴보고, 제 3장에서는 검색을 위한 알고리즘과 시스템, 프로토타입의 구조에 대해서 소개하겠다. 제 4장에서는 프로토타입을 평가해보고 향후 과제를 정리하겠다.

2. 관련 연구

2.1 블로그 검색

블로그라는 개념이 새로운 정보 검색의 원천으로 사람들의 주목을 받기 시작하면서(Fujiki, T., Nanno, T., Suzuki, Y. and Okumura, M, 2004) 블로그만을 대상으로 하는 다양한 검색 방법에 대하여 논하고 있는 논문도 등장한다(강순희, 2008). 이러한 블로그만을 대상으로 검색하는 논문들 중에서 키워드들을 적절히 추출하여 비교하는 방식으로 검색을 진행한 논문들도 있는데, 이 논문에서는 우선 해당 텍스트에서 키워드들을 뽑아내어 카테고리를 만들고 그렇게 하여 만들어진 도메인들을

중심으로 검색된 결과들을 분류하여 좀 더 사용자가 의도한 도메인에 맞는 정보를 주고자 하고 있다(Mishne. G., and de Rijke. M, 2006). 본 논문과 같은 맥락의 문제 의식을 가지고 시작한 분류 방법이라고 할 수 있겠다. 또한 한 번 사용자가 입력한 쿼리를 중심으로 키워드 맵을 작성하고, 그 중에서 사용자가 원하는 키워드를 골라서 설정하게 되면, 다시 그 설정된 키워드를 가지고 검사를 수행하는 방법도 소개되고 있다(Takama, Y., Kajinami, T., and Matsumura, A, 2005). 일종의 피드벡 기능을 가지고 사용자가 자신이 원하는 영역에 도달할 수 있도록 도와주는 검색 엔진이다. 주제, 블로거(블로그에 글을 올린 사람), 블로그의 평판을 중심으로 가중치를 두어 블로그 검색을 차별화 하겠다는 논문도 있었다(Toda. Hiroyuki., Fujimura. Ko., Inoue. Takafumi., Hiroshima. Nobuaki., Sugizaki. Masayuki, and Kataoka, 2007).

2.2 형태소 분석과 클러스터링

형태소 분석은 자연어로 적혀있는 언어 속에서 그 기본형을 찾아내는 것이며, 어간, 어미 등을 가지는 한국어 고유의 특성 때문에 필요하다. 한국어 문서에서 이것이 무엇에 관한 글인지 알기 위해서는 각 단어의 기본형을 추출해와야만 하는데 그 과정 자체 만으로도 논문이 나오고 있다. 이런 한글의 특성에 기초하여 키워드를 추출해 내는 형태소 분석에 대한 논문이 몇 편 있으며(김재한, 안미정, 옥철영, 1993; 김재한, 옥철영, 1994), 이런 형태소를 분석하는 이론들을 조사하며 키워드들을 분류해 냈다. 실제로 많은 검색엔진들이 이런 형태소 분석기를 사용하여 한국어 텍스트들을 분류하고 있었다(이은철, 김성천, 황영섭, 1992).

이렇게 분류된 키워드들의 어느 정도까지를 유사 자료로 취급할 수 있는지에 대한 기준 또한 굉장히 중요한 문제이다. 키워드들을 두고, 전체의 제목과 비교하여 두 문헌이 유사하다고 할 수 있는지 없는지에 대한 기준은 상황에 따라 여러 가지가 존재하고 있으며 연구도 진행되고 있다(정영미, 2005). 여기서 문헌 클러스터링 기법으로 제시되고 있는 여러가지 계수들을 살펴보며 효과적으로 유사 자료를 연결할 수 있고 실행 시간도 최소화 할 수 있는 식을 찾았다.

3. 알고리즘와 시스템

3.1 Problem Solving

궁극적으로 검색사이트들이 검색결과의 분류를 하고 검색 방식을 만드는 이유는, 최종사용자가 검색을 했을 때 자신이 의도한 검색결과를 쉽게 얻을 수 있도록 하기 위함이다. 현재 네어버 같은 검색포털들도 카테고리를 통한 검색 결과를 제공하고 있고, 기본적으로 어떠한 검색을 하면, 이미지, 동영상, 블로그, 카페, 사전 등의 항목으로 좀 더 사용자가 자신이 원하는 정보를 찾아갈 수 있도록 지원해 주고 있다. 그러나 점점 다양해지고 세분화 되며 변종들이 많이 생기는 현재의 정보들 속에서, 이러한 의도된 카테고리 중심으로 모든 정보들을 분류시켜 넣는 것은 한계가 있다. 사용자들은 블로그에 자신이 원하는 대로 다양한 컨텐츠들을 등록하고 있으며, 수많은 주제들이 존재하기 때문에 각 키워드들은 전혀 다른 형태의 도메인을 가지게 된다. 가장 이상적인 검색엔진이라면 사용자가 의도한 영역의 정보들만 수집하였다가 제시하면 되지만, 웹상에서 사용자의 개인적인 취향을 일일이 맞추어 주는 것은 불가능하고, 각각의 검색에 다른 영역의 정보를 원

하는 경우도 있을 수 있으며, 수많은 분류 영역 속에서 해당 블로그를 어느 쪽으로 분류하여 놓을 것인지도 마땅치 않다. 잘못하면 사용자를 위한다고 편집하여 놓은 검색 결과가 사용자가 실제로 찾아보고자 했던 정보를 삭제하고 보여 줄 수도 있는 것이다. 따라서 일단은 해당 검색어가 가지고 있는 컨텐츠들을 아무런 틀도 가지지 않고 분석하는 것이 다양한 사용자의 요구를 벗어나지 않을 것이다. 그리고 그에 따라서 이 검색어에 해당하는 유명 주제들은 어떤 것들이 있었는지 모두 알려주고, 전체 자료에서 비중은 어떠했는지 함께 표시하는 것이 한 눈에 해당 검색어의 추세를 살펴볼 수 있을 것이다.

논문 제 1장에서 제시한 현상들로부터 발생하는 문제점을 인식하고 이에 따른 개선사항들을 정리하여 보면 다음과 같다.

- 똑 같은 키워드를 모두 담고 있다 할지라도 종류별로 관계 있는 자료들끼리 분류하여 볼 수 있게 해주는 시스템.
- 자료마다 대중의 선호도, 유명도, 유행도, 필수도 등의 요소들로 순위를 매겨 한 눈에 알기 쉽게 보여주는 시스템.

이번 클러스터링을 통해서 추구하는 결과는 블로그 검색 속에서의 좀 더 효과적인 결과를 제시하기 위함이다. 논문 제 1장에서의 예를 들면, '에쿠니 가오리'라는 소설가에 대하여 검색을 시행했을 때, 대중들이 그녀에 관해 가장 많이 포스팅한 정보가 맨 위에 모아져서 제시되는 것이며, 그것에 이어서 다음으로 많이 포스팅 된 같은 주제에 대한 정보들이 순차적으로 나타나는 것이다. 다음으로 역시 '유이'란 검색어를 입력하였을 때, 동명이인의 자료들은 따로 분류가 되는 것이고, 나아가 배우인 사람을 검색하면 출연분야나 작품별로 분류가 가능하며, 가수인 사람을 검색하면 곡 별로 따로 분류가 가능하게 되는 것이다.

이 프로토타입 시스템은 네이버 블로그의 자료들을 중심으로 1차적으로 제작되었다. 블로그들의 제목들을 모아오고, 제목들 속에서 의미를 가지는 형태소들을 분류해 내었다. 그 후에 서로의 유사도를 측정하는 알고리즘을 통하여 관련성이 높다고 여겨지는 자료들을 한꺼번에 묶어서 그룹화 하였으며, 모든 작업이 끝난 후에 가장 규모가 큰 그룹의 순서로 순위를 매겨 제시하였다.

3.2 Definition of Key Algorithm

Definition 1 : 인터넷으로부터 검색을 요청, 수행한 뒤 결과로부터 추출한 제목들을 T로 정의 한다. 따라서 $T = \{T_1, T_2, ..., T_n\}$로 표현되고 n개의 제목을 가지고 있음을 의미한다.

Definition 2 : 각 제목으로부터 형태소 분리를 통해 얻은 하나하나의 단어, 키워드들을 First Keyword Set FK로 정의한다. 따라서 $FK = \{FK_{ij} \mid i = 1, 2, \cdots, n, j = 1, 2, \cdots, m\}$로 표현되고 n개의 T가 있었다면 i도 물론 n개일 것이다. j는 각 제목이 가지고 있는 키워드를 나타내며 임의로 m개를 가지고 있음을 뜻한다.

Definition 3 : 각 제목들이 가지고 있는 키워드의 개수를 W로 정의한다. 그런데, 각각의 키워드는 검색어를 포함하고 있으므로, 검색어를 제외한 나머지 키워드의 개수를 Weight로 가지게 된다. 따라서 $W = \{W_i = |$

K_{ij}|-1 | i = 1, 2, ⋯, n}로 표현된다.

Definition 4 : FK를 W에 따라 정렬하여 만든 Keyword Set을 K로 정의한다. K = {K_{ij} | i = 1, 2, ⋯, n, j = 1, 2, ⋯, m}로 표현되고 i는 n의 수와 같다.

Definition 5 : 일치도 검사 수행시, 다른 K들이 자신의 일치성을 검사 할, 현재의 중심 K를 K_x로 정의한다. 따라서 $K_x \in K$로 정의된다. 모든 일치도 검사는 임의의 K_x를 하나 두고 다른 K들이 이 K_x와 일치도를 검사하는 과정으로 이루어진다.

Definition 6 : 일치도 검사를 통해, 관련도가 높다고 인식되는 자료들은 Result Set에 저장되게 된다. 이것은 기호로 Res = {Res_p | p = 1, 2, ⋯}로 정의된다. 또한 각각의 Res_p는 Res_p = {T_q | $T_q \in T$}로 정의 된다. 여기서의 q는 그저 임의의 q∈n인 숫자를 나타낸다. 즉, 하나하나의 Res_p은 각각의 Title을 임의의 수만큼 원소로 가지고 있는 것이며, 이런 Res_p의 집합이 최종의 Result Set을 구성하게 된다. 그리고 P는 결과 set의 개수가 된다.

3.3 Preparing Process

형태소 분리 작업을 거쳐 키워드를 분리해 내는 작업을 끝마쳤다면, 두 제목을 비교하는 방법에 대한 고민을 하게 되며 이 부분이 핵심적인 요소라고 할 수 있겠다. 중복을 최소화하면서 효과적으로 두 제목을 비교해 나가기 위해, 키워드 Set의 K_{ij}를 j의 개수(W)에 따라 정렬하는 방법과 서치 리스트를 만들어 쓸데없는 비교를 하지 않는 것을 생각하게 되었다.

우선적으로 FK에서 키워드를 가장 적게 가지고 있는 제목들을 우선적으로 맨 앞쪽에 위치시키기 위한 목적으로 K를 만든다. 이런 과정을 거치게 되면 결국 W값이 가장 작은, 짧은 길이의 제목들이 맨 위에 들어가게 되게 먼저 일치도 여부를 검사하는 대상이 되게 된다. 위와 같이 키워드 Set을 만드는 정렬과정은 일반적인 정렬 알고리즘으로 W의 크기에 따라 정렬하는 것으로 가능하다.

서치 리스트는 단순히 키워드들의 flag정보가 들어있는 리스트이다. Flag의 초기 값은 모두 참으로 되어있으며, 검색을 진행해 나감에 따라 키워드가 K_x가 될 대상인지 아닌지를 판단하여, K_x의 대상이면 참값을, 대상이 아니면 거짓값을 가지고 있다.

일치도 검사의 시작은 서치 리스트에서 비교의 중심이 될 K_x를 하나 가지고 오는 과정에서 시작된다. 이 K_x를 가지고 오는 과정은 순차적으로 이루어지기 때문에, 아직 일치도 검사의 중심어가 되지 않은 K들 중에서 W가 가장 낮은 것부터 차례차례 가져오게 된다. 따라서 실제로 일치도 검사의 과정은 W가 제일 낮은 K부터 높은 K까지로의 과정을 거치며 이루어진다고 할 수 있겠다. 이렇게 중심어로 선택된 K_x를 대상으로 하여 키워드 Set 안의 모든 K들이 일치도 검사를 수행하게 되는데, 일치도 검사 공식에 합격하고 몇 가지 예외 경우에 해당하지 않게 되어, 관련자료로 인식되게 되면 Result Set으로 들어가게 된다. 이런 방식으로 어떤 자료와 충분히 일치했다고 여겨지는 자료는 Result Set으로 들어가는 동시에 서치 리스트에서 제외된다.

3.4 Code of Algorithm

<그림 1>의 코드는 일치도 검사를 하는 핵심과

정을 보여주는 알고리즘이다. 여기에 등장하는 각각의 함수에 대한 설명은 아래와 같다.

combine() : 일치도 검사를 통해 Result Set을 만드는 과정 전체를 수행하는 함수.

search_in_searchList() : 서치 리스트속에서 다음에 검색의 중심이 될 K_x를 찾는 함수.

remove_from_searchList() : 함수는 서치 리스트속에서 해당 K의 flag를 false로 만들어 K_x의 대상에서 제외시키는 함수.

Combine 함수는 자신의 내부에서 자신을 다시 호출하는 재귀함수의 형태로, 서치 리스트의 모든 flag가 거짓으로 설정되어 더 이상 K_x를 만들 수 없을 때까지 반복하게 된다.

검색을 할 때에 고려해주는 예외사항들을 <그림 2>를 통해 몇 가지 확인할 수 있다. 먼저는 어떠한 키워드 K_{ij}의 길이가 2byte미만의 한 글자 정도의 크기라면, 카테고리 분류를 하는 게 거의 의미가 없는 글자로 판단하고 비교를 하지 않았다. 그리고 이렇게 제외되는 글자들은 모두 뒤에서 일치도 여부를 계산할 때도 제외되게 된다. 그리고 weight를 너무 적게 가지고 있는 K들도 검색에서 제외하였는데, 이것은 해당 검색어를 입력하게 되면 응당 나오게 되어있는 당연한 하나의 관련어(關聯語) 때문에 수많은 자료들이 하나의 카테고리로 묶이는 것을 방지하기 위함이다. 하지만 이런 경우, 비록 weight가 낮지만 완벽히 일치하는 제목의 자료들을 묶기 위해, 두 개의 weight가 같은 경우에는 일치도 검사를 한다.

최종적으로 형태소 기반 일치도 검사를 하는 식은 다음과 같다. a라는 값이 일종의 상관계수가 되는 것이다.

```
p //결과 그룹의 수를 정의할 p를 정의
combine()
{
    x = search_in_searchList();      //비교의 중심이 될 키워드를 찾아 변수 x에 둔다.
    num_cmp = 0;                     //일치하는 키워드의 수가 몇개나 있는지 저장한다.
    except = 0;                      //검색어와 일치하는 형태소는 제거한다.

    if(x != -1)                      //더 이상 검사할 K가 있는지 알아본다
    {
        Wx = |Kx|-1;                 //keyword를 뺀 weight를 입력
        p++;
        insert(Tx, Res[p])           //현재 타이틀을 result set에 넣는다
        remover_from_searchList(Tx)

        for(int y = 0; y < n; y++)
        {
            if( x != y)              //중심키워드와 겹치지 않는 키워드이면 비교.
            {
                Wy = |Ky|-1;  //중심키워드와 비교 할 각각 키워드의 weight를 구한다

                for(int i = 0; i < |Kx|; i++)
                {
                    //키워드가 한 글자인 것은 의미가 없으므로 무시한다.
                    if((Wx > 2) || (Wx == Wy) || (|Kxi| > 2))
                    {
                        if(Kxi != 검색어)
                        {
                            for(int j = 0; j < |Ky|; j++)
                            {
                                //두 키워드가 일치하면 일치도를 높인다.
                                if(Kyj != 검색어 && |Kyj| > 2)
                                {
                                    if(Kxi == Kyj)
                                    {
                                        num_cmp++;
                                    }
                                }
                            }
                        }
                        else  //해당 형태소가 검색어였다면 weight에서 제외한다.
                        {
                            except++;
                        }
                    }
                }
                if((num_cmp/(Wx-except)) > a)  //일치도 검사 공식에 적용한다.
                {
                    insert(Ty, Res[p])  //현재 타이틀을 result set에 넣는다.
                }
                num_cmp=0;
                except=0;
            }
        }
        combine();
    }
}
```

<그림 1> 일치도 검사 알고리즘

$$\alpha = \frac{K_x\text{일치하는 키워드의 수}}{K\text{의 } weight - K\text{의 } except}$$

a는 위와 같이 이루어 지게 되는데, 이 식이 일정 수치를 넘어서면 같은 자료로 인식을 하게 된다. <그림 2>에서는 이 값을 a(threshold)로 설정하고 있는데, 세세한 분류를 하고 싶다면 이 값을 더 높게 설정하면 되고, 어느 정도 비슷한 영역에 있는 것 만으로 같은 자료로 분류하고 싶으면 이 값을 더 낮게 설정하면 된다. 단순하고 효율적인

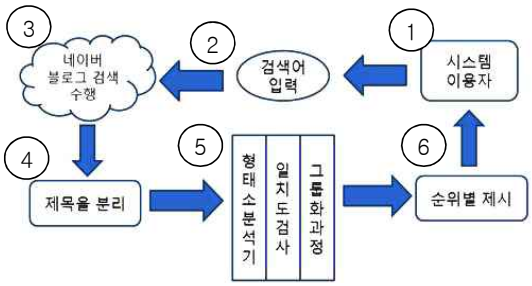
일치도 검사 방법인 만큼 역시 추후 더욱 연구하여 이상적인 일치도 검사 기법을 찾을 계획이다.

3.5 Result for user

Result Set을 구하는 분류작업이 다 끝나게 되면, 순위를 매기기 위한 또 한번의 정렬작업을 하게 된다. 이 과정은 $|Res_p|$의 크기를 중심으로 이루어지는데 가장 많은 수의 결과를 가지고 있는 Res_p을 맨 앞으로, 다음으로 많은 수의 결과를 가지고 있는 Res_p를 다음으로 제시한다. 역시 이 과정도 퀵 소트를 이용하여 진행하였으며 실행시간은 O(nlogn)이 된다. 사용자에게 전해 지는 형태는 <예 1>에서 제시된 것과 같은 결과 형태를 띄게 되며 순위와 퍼센트가 존재하게 된다. 이 퍼센트는 $|Res_p|$를 전체 제목의 개수 n으로 나누어 줌으로써 쉽게 구할 수 있는데, 이것을 제시하는 것은 매우 중요하다. 이 퍼센트를 통해서 사용자는 블로그 상에 해당 키워드에 대한 정보가 중요하고 유명한 몇 가지 키워드에 의해 독점당하는 성향인지, 여러 가지 다양한 키워드들에 의해 쭉 퍼져있는 성향인지를 파악할 수 있게 된다. 따라서 먼저 나온 자료의 퍼센트가 높다면 사용자는 앞쪽에 있는 몇 개의 자료들을 중심으로 검색어를 파악할 것이며, 그 퍼센트가 낮다면 사용자 뒤쪽에 제시되는 정보들까지 세세하게 살펴보게 될 것이다.

3.6 System Architecture

<그림 2>는 본 논문에서 제안하고 있는 블로그 검색시스템의 구조도를 보여주고 있다. 우선, 시스템의 사용자가 검색어를 입력하게 되면 그 검색어를 가지고 네이버의 블로그 검색을 수행 한 후 결과 중에서 제목만을 분리해 내어 저장하게 되는데, 이 제목들이 클러스터링을 위한 문헌요소로 사용



<그림 2> System architecture

된다. 다음으로 제목들은 형태소 분석기를 거쳐, 각각의 기본형의 단어를 가지게 되는 키워드의 형태를 갖추게 된다. 이 키워드들의 일치도 검사를 통해 두 자료가 같은 자료로 묶을 것인지에 대한 판단을 하게 되는데, 이 일치도 검사의 알고리즘은 뒤에 자세히 설명하겠다. 모든 자료들을 비교하여 그룹화 과정을 마치게 되면, 가장 자료수가 많은 그룹을 맨 앞에 위치시키는 정렬과정을 거치게 된다. 정렬과정을 끝마치게 되면 순위가 매겨지고, 전체 자료의 수 중에서 해당 그룹에 할당된 자료의 비율이 동반되어 시스템 이용자에게 결과로 제공된다. 실질적으로 형태소 분석기는 여러가지 논문과, 한글 검색과정에서 평범하게 사용되는 것이기 때문에 자체적으로 깊은 연구를 하지는 않았다. 그러나 블로그의 특성상 각종 이모티콘이 등장하였으며 그것을 분리하는 작업을 또 수행하였다. 그룹화 과정은 퀵소트 같은 널리 알려진 정렬 알고리즘을 사용하였으며, 검색의 효용성의 측면에서 큰 영향을 끼치지 않는다. 왜냐하면 이 과정은 실행시간에는 O(nlogn)이라는 영향을 끼치지만, 검색어를 분류, 관련도를 확인하는 과정과는 별반 상관이 없기 때문이다. 그러므로 결국에는, 일치도 검사를 하는 알고리즘을 얼마나 효과적으로, 또 유기적으로 제작하여 제시하느냐가 이 논문의 유용성을 판단하는 핵심 척도가 된다고 할 수 있겠다.

3.7 Specific example

한 예로 다음과 같은 검색어가 있다고 가정하자.

<예> '메탈리카'가 검색어로 제시된 상황. 만약 이 키워드에 잘 모르는 사람이 단순히 검색을 시행하였다면, 30년 가까이 되도록 활동해 온 이 그룹의 주요사항을 파악하는데 꽤나 많은 시간이 걸릴 것이다(원속의 숫자들은 <그림 2>의 과정과 각각 연결된다).

➜ ①, ② : 사용자가 '메탈리카'라는 쿼리를 전송
➜ ③ : 네이버에서 블로그 검색을 시행
➜ ④ : 네이버 검색결과의 제목을 분리
1. 메탈리카! 내한 공연 임박
2. 메탈리카 좋아하는 노래 one
3. 메탈리카 내한 공연 후기
4. 2006 메탈리카 내한 공연 관람기
5. 우리집 메탈리카들

➜ ⑤ : 형태소 분석기로 분류
1. [우리집][메탈리카]
2. [메탈리카][내한][공연][후기]
3. [메탈리카] [내한][공연][임박]
4. [메탈리카][좋아하는][노래][one]
5. [2006][메탈리카][내한][공연][관람기]

➜ ⑤ : 일치도 검사
- 1과 2, 3, 4, 5번은 키워드 '메탈리카'를 제외하고 일치하는 문자가 없다.
- 2와 3, 5번은 키워드를 제외한 단어들 중에 [내한], [공연]이라는 2개의 단어들이 서로 일치한다. 이를(일치 키워드/전체 키워드)의 수식으로 표현하면 각각 2/3, 2/3, 2/4의 일치성을 갖는다고 볼 수 있다.
- 4와 1, 2, 3, 5번은 키워드 '메탈리카'를 제외하고 일치하는 문자가 없다.

➜ ⑤ : 그룹화 과정
- 2, 3, 5번 제목을 한 그룹으로 인식한다.
- 1번 제목을 한 그룹으로 인식한다.
- 4번 제목을 한 그룹으로 인식한다.

➜ ⑥ : 순위별 분류 작업을 시행
➜ ⑥ : 순위별로 제시된 결과
1. 검색어 : 메탈리카, 내한, 공연 60%
   - 메탈리카! 내한 공연 임박
   - 메탈리카 내한 공연 후기
   - 2006 메탈리카 내한 공연 관람기
2. 검색어 : 메탈리카, 좋아하는, 노래 one 20%
   - 메탈리카 좋아하는 노래 one
3. 검색어 : 메탈리카, 우리집 20%
   - 우리집 메탈리카들

3.6.1 Example of using keyword 'metallica'

시스템의 이용자가 맨 마지막에 얻게 되는 결과는 3.2.1.의 순위별로 제시된 결과이다. 사용자는 이것을 통해 메탈리카가 한국에 내한을 왔었기 때문에 그에 관한 포스트가 요즘 주를 이루고 있으며 전체적인 비중은 얼마인지 알 수 있다. 물론 여기서 내한 공연 임박과 내한 공연 후기는 큰 상관이 없다고 느낄 수도 있다. 이것은 일단 메탈리카라는 밴드의 전체를 조명한 것이기에 위와 같은 결과가 나오게 되지만, 내한에 대한 좀 더 상세한 결과를 보고자 한다면, '메탈리카 내한'이라는 키워드로 추가 검색을 할 수 있을 것이다. 두 번째는 one이라는 노래에 관한 포스팅이 주를 이루게 될 것이고, 세 번째 결과는 메탈리카라는 밴드에 대한

이야기가 아닌 장식품에 대한 이야기이다. 우리집, 장식품 등에 관한 이야기는 메탈리카라는 검색어에 비추어 봤을 때, 상당히 소수에 지나지 않기에 한꺼번에 묶여서 뒷부분에 위치하게 될 것이다. 따라서 사용자는 메탈리카라는 단어에 대해서 블로그의 추세를 살 필 때, 장식품은 큰 비중이 없음을 알게 될 것이고, 별로 중요하지 않은 정보가 앞 쪽에 나타나는 것을 방지하여 좀 더 체계적인 정보를 얻을 수 있을 것이다. -실제로 현재 네이버 블로그에서 메탈리카를 검색하게 되면 장식품인 메탈리카 앞 페이지에 상당 수 나타난다.-이를 통해 사용자는 앞에서 의도한 대로, 해당 검색어에 대해 블로그에 포스팅되어 있는 자료의 추세를 살펴보기 용이할 것이다.

4. 평가

위에 제시한 프로토타입을 구현하여 실제로 블로그 검색을 시행하여 보았다. 검색 엔진은 네이버를 사용하였으며, 검색 자료는 네이버 블로그 검색 결과의 맨 앞 페이지부터 30페이지까지 300개의 블로그를 대상으로 하였다. a의 값은 0.2, 0.4, 0.6, 0.8 4가지로 설정하고 각각의 경우에 대하여 시험해 보았다. a의 값은 일치도를 평가하는 숫자로 3.4에 그 값을 구하는 방법이 나와 있으며, 이 값이 낮으면 좀 더 일치도가 낮은 블로그들도 하나의 주제로 묶겠다는 의미가 되며 이 값이 크다면 일치도가 아주 높은 자료들만 한 주제로 묶겠다는 의미가 된다. 결과는 3가지 값을 구하는 것으로 하였다. 첫째는 클러스터링한 그룹의 개수를 통해 얼마나 많은 주제들로 분류가 되었는지 살펴보았고, 둘째로 그룹 당 평균 블로그 수를 구하여 하나의 주제에 어느 정도의 블로그가 들어있는지를 알아본다. 그런데 단순히 평균만을 통해서는 유명한 자료들은 어느 정도로 분포되어 있는지 알 수 없기 때문에 분산을 구하여 첨부해 보았다.

<표 1> '에쿠니 가오리'로 검색한 결과

	0.8	0.6	0.4	0.2
그룹의 개수	83	83	81	81
그룹 당 평균 블로그 수	3.61	3.61	3.70	3.70
그룹 당 블로그 수의 분산	38.98	38.90	39.56	39.50

<표 2> '아라가키 유이'로 검색한 결과

	0.8	0.6	0.4	0.2
그룹의 개수	78	73	66	61
그룹 당 평균 블로그 수	3.84	4.10	4.54	4.91
그룹 당 블로그 수의 분산	102.84	108.83	121.18	129.32

위의 <표 1>과 <표 2>에서처럼 '에쿠니 가오리'라는 소설가와 '아라가키 유이'라는 배우에 대하여 검색을 시행하였다.

소설가 '에쿠니 가오리'는 그녀의 유명한 몇 가지 작품 외에 크게 여러 분야에 걸쳐서 활동하지는 않기 때문에 다양한 종류의 블로그가 존재하지는 않았다. 애초에 책 제목이 일치하는 정도면 a이 낮으나 높으나 함께 모였고, 그렇지 않으면 어떤 상황에서나 다른 자료로 인식 되었다. 따라서 a값을 달리하여도 크게 달라지지 않는 양상을 보였다. 유명한 저서에 관해서 착실하게 분류가 진행되었으며, 단순히 키워드 '에쿠니 가오리'만을 제목으로 한 블로그가 상당히 많았는데, 여기서 키워드만을 포함한 제목은 특별히 분류하지 않았기 때문에 하나짜리 그룹의 수가 늘어나 그룹의 개수는 늘어

나게 되었다.

배우 '아라가키 유이'의 경우는 드라마 외에도 영화, CM 등의 여러 분야에 걸쳐 활동하고 있기 때문에 a값을 조금씩 달리할 때마다 다른 결과가 나왔다. 그러나, 모든 작품에 주연을 하거나 히트를 친 것은 아니라 유명한 작품을 위주로 분산이 매우 크게 나왔음을 알 수 있다.

다음으로는 클러스터링된 그룹들이 얼마나 관련도 있나를 평가하기 위해 Clustering SIMilarity (CSIM)을 사용하여 프로토타입을 평가하여 보았다(Chung, Y.M., Lee, J.Y., 2001). CSIM은 클러스터링의 유효성을 평가하는데 자주 쓰이는 Rand's Method(Rand, W.M., 1971)에서 고안된 것으로 Rand's Method의 단점을 극복하기 위해 다이스의 계수를 적용한 것이다. 여기서 우리는 현시점의 네이버 블로그 데이터 셋 50개를 기준으로 블로그 자료에 대한 어떠한 가공도 없이 시행하였다. 비록 우리가 만든 프로토타입이 모든 웹 블로그를 대상으로 실험을 해볼 수 있지만 CSIM로 측정하는데 있어서 시간상의 효율을 고려해 일단 50개의 페이지를 모델로 평가하여 보았고 추후에 좀 더 자세한 평가를 할 예정이다. 프로토타입이 블로그 클러스터링을 하기 전에, 미리 수동으로 50개의 블로그를 클러스터링 하고 프로토타입이 그것의 결과물을 만들어내면 CSIM 방법으로 평가를 실행하였다.

<표 3> CSIM으로 평가한 결과

검색어	에쿠니 가오리	아라가키 유이
CSIM Value	0.934	0.761

CSIM Value가 1에 가까울수록 이상적인 분류를 하였다고 판단되는데 2개의 수치가 0.934, 0.761로 높은 수치를 보여 성공적인 클러스터링을 해냈다고 평가할 수 있다. 해당 검색어에 대해 많은 키워드를 가지는 아라가키 유이의 검색어 쪽이 변수를 더 많이 가지기 때문에 비교적 낮은 CSIM 결과값을 가졌다.

평균과 분산, CSIM등의 방법으로 평가해 본 결과 이러한 검색을 통하여 사용자들은 해당 검색어에 가장 많이 들어있는 내용과 전체적인 흐름을 쉽게 알 수 있을 것으로 기대되며, 개인 사용자 이외의 기업 같은 곳에서도 해당 검색어를 입력했을 때 사람들이 자신의 상품에 대하여 어떠한 곳에 가장 관심을 두고 있는지 살펴볼 수 있을 것이다.

5. 결론

블로그 그룹화를 추구하는데 있어서 첫째의 평가요소는 이 알고리즘이 얼마나 사용자들이 보기 편하게 정보를 제공하느냐에 있다. 따라서 일치도 검사를 하는데 있어서 어느 선까지를 같은 주제로 인식할 것인지, 유사도 레벨을 어떻게 설정할 것인지의 여부가 이 연구를 진행하는데 있어서 가장 큰 사안이 된다고 할 수 있겠다.

지금은 위에서 제시한 일치도 식에 따라서 간편하게 프로토타입을 구현하여 보았지만, 이 이상으로 실험해 볼만한 클러스터링 식들도 많이 연구되어 있고(권오병, 문예성, 김민용, 2007), 이 외에 다양한 유사도 비교 방법을 연구 중에 있다. 이러한 다양한 공식들을 가지고 다양한 클러스터링을 시도하여 가장 사용자에게 의도한 정보를 잘 전달할 수 있도록 비교, 분석, 발표할 예정이다. 그리고 추후 문헌의 유사도의 레벨을 사용자가 선택하도록 하여 자신이 원하는 선에서 그룹화를 하도록 하는 '유사도 레벨의 옵션화'에 대한 연구도 필요하다.

다음으로는 한글이라는 문자의 특성을 잘 살려주는 자연어 처리, 키워드 처리에 대한 연구이다.

궁극적으로 제목을 뽑아내어서 키워드를 만들 때 검색에 도움이 될 최적의 조건으로 키워드를 선별해 내는데 그 목적이 있다. 형태소 처리기를 사용하여 해당 한글의 기본형을 추출하는 것 이외에도 영어나 한자가 섞여있는 예외적인 경우도 많이 등장한다. 이런 경우들 중의 문제는 해당 검색어를 그저 영어로 한 번 바꾸어 쓴 제목들도 빈번히 등장하는데, 검색어와 다름없는 키워드임에도 불구하고 이 키워드를 중심으로 관련자료를 묶게 되면 아무 관련이 없는 자료들이 무더기로 모여버리고 만다. 이런 경우를 대비해 한국어로 어떤 검색어를 입력하였을 때, 그것과 정확하게 대응하는 타국의 언어를 키워드에서 제외하는 시스템을 개발하여 좀 더 좋은 결과를 제시하고자 한다.

그리고 위의 자연어 처리와는 다르게 동음이의어나 동의어에 대한 처리도 여전히 연구대상이다.

지금까지 주제 유사성을 기반으로 클러스터링을 하여 결과를 제시하는 블로그 검색 방법을 제안하고 프로토타입을 만들어 실험을 해보았다. 실험 결과로 제시 받은 결과들을 살펴보면 300개의 블로그 자료들이 유명한 자료들을 중심으로 묶이는 것을 확인할 수 있었으며, 여러 사용자들이 이용하여 해당 검색어의 추세를 살펴보는데 도움이 될 것이라고 생각한다. 위에 제시된 추후 연구사항을 토대로 연구를 진행할 것이며, 전체적으로 블로그와 블로그 검색이라는 분야에 대해 앞으로 더욱 많은 연구가 필요할 것이라고 생각한다.

참고문헌

강순희, “블로그 검색에 대한 연구”, 한국정보관리학회, *한국정보관리학회 학술대회논문집*, 2006.

김재한, 옥철영, “통합 형태소를 이용한 한국어 형태소 분석기”, 한국정보과학회, *한국정보과학회 학술발표논문집 한국정보과학회 1994년도 가을 학술발표논문집*, 21권 2호(A), 1994. 10.

김재한, 안미정, 옥철영, “활용 형태소에 기반한 한국어 형태소 분석기”, 한국정보과학회, 한국정보과학회 학술발표논문집, *한국정보과학회 1993년도 가을 학술발표논문집*, 20권 2호(1993).

김중태, “나는 블로그가 좋다”, 서울 : 이비컴, 2004.

권오병, 문예성, 김민용, “블로그 사용에 영향을 미치는 요인” 한국전자거래학회, *한국전자거래학회지*, 12권 4호(2007).

이은철, 김성천, 황영섭, 이종혁, “한국어 텍스트 검색 시스템 KIRT의 구현”, 한국정보과학회, 한국정보과학회 학술발표논문집, *한국정보과학회 1992년도 봄 학술발표논문집*, 19권 1호(1992).

정영미, “문헌 클러스터링”, 정보검색 연구 구미무역㈜ 출판부, 2005.

최호찬, “[요즈음 문화] 인터넷의 새로운 문화 블로그-개인이 인터넷에서 존재하고 대화하는 방법”, *경향잡지*, 2004년 3월호(통권 1632호).

Chung, Y. M. and J. Y. Lee, “A corpus-based approach to comparative evaluation of statistical term association measures”, *In J. of the American Society for Information Science and Technology*, 2001.

Fujiki, T., T. Nanno, Y. Suzuki, and M. Okumura, “Identification of Bursts in a Document Stream”, *First International Workshop on Knowledge Discovery in Data, Streams*, 2004.

Mishne, G., M. de Rijke, “A Study of Blog Search”, *Lecture notes in computer science*, Vol.3936 (2006).

Rand, W. M., “Objective Criteria for The Evaluation of clustering Methods”, *In J. of the American Statistical Association*, 1971.

Takama, Y., T. Kajinami, and A. Matsumura, “Blog Search with Keyword Map-Based

Relevance Feedback", *Lecture notes in computer science*, Vol.3614(2005).

Tim O'reilly., web 2.0 conference in Sanfrancisco USA, 2004.

Toda, Hiroyuki., Fujimura, Ko., Inoue, Takafumi., Hiroshima, Nobuaki., Sugizaki, Masayuki., Kataoka., "BLOGRANGER : Implementation of Goal-oriented Blog Search Engine", *Transactions of Information Processing Society of Japan*, Vol.48(2007).

Abstract

Study for Blog Clustering Method Based on Similarity of Titles

Ki Jun Lee* · Myung Jin Lee* · Woo Ju Kim*

With an exponential growth of blogs, lots of important data have appeared on blogs. However, since main topics mentioned in blog pages are quite different from general web pages, there are problems which can't be solved by general search engines. Therefore, many researchers have studied searching methods only for blogs to help users who want to have useful information on blog. We also present a blog classifying method based on similarity of titles. First, we analyze blogs and blog search engines to find problems and solution of current blog search. Second, applying our similarity algorithm on blog titles, we discuss a way to develop clustering method only for blog. Finally, by making a prototype system of our algorithm, we evaluate our algorithm's effectiveness and show conclusion and future work. We expect this algorithm could add its power to current search engine.

Key Words : Search, Blog Search, Clustering, Blog Clustering

* Information and Industrial Engineering, Yonsei University

저 자 소 개

이기준

2008년 연세대학교 컴퓨터산업공학과 학사를 취득하고, 2008년부터 현재까지 연세대학교 대학원 정보산업공학과 석사과정에 재학 중이다. 관심분야는 웹 서비스, 유비쿼터스 웹 서비스, 검색 엔진, 시멘틱 웹 서비스이다.

이명진

현재 연세대학교 정보산업공학과 지능정보시스템 연구실 박사 과정에 재학 중에 있으며, 다수의 XML 및 XML Web Services 서적을 집필하였다. 관심분야는 시맨틱 웹과 의미기반 검색, 시맨틱 웹 포털, 시맨틱 웹 서비스이다.

김우주

1987년 연세대학교 BBA 과정 학사 학위를 취득하고, 1994년 KAIST 경영과학 박사를 취득하였으며, 현재 연세대학교 정보산업공학과 교수로 재직 중이다. 관심분야는 시맨틱 웹, 시맨틱 웹 환경의 의사결정지원 시스템, 시맨틱 웹 마이닝, 지식관리 및 인공지능 웹 서비스이다.