한국지능시스템학회 논문지 2011, Vol. 21, No. 6, pp. 724-729
http://dx.doi.org/JKIIS.2011.21.6.724

사용자 검색 질의 단어의 순서 및 단어간의 인접 관계에 기반한 검색 기법의 구현

Implementation of Search Method based on Sequence and Adjacency Relationship of User Query

소병철[1] · 정진우[1,+]
Byung-Chul So and Jin-Woo Jung

[1]동국대학교 컴퓨터공학과

요 약

정보 검색은 다수 자료에서 사용자가 원하는 부분을 찾는 과정을 의미한다. 일반적으로 대규모 자료 집합의 관리를 위해서는 데이터베이스가 사용되는데 인터넷과 같은 복잡한 문서구조들이 공존하는 환경에서는 한 번에 사용자가 원하는 문서를 정확히 찾아내는 것이 어렵기 때문에, 문서에 순위를 부여하여 사용자에게 제시하는 방법이 일반적으로 많이 사용된다. 본 논문에서는 자료에 포함되어 있는 단어들을 단순히 검색하는 것 뿐만 아니라 단어들 간의 순서 및 인접성을 고려한 검색방법을 용어빈도-역문헌빈도 및 n-gram 기법을 응용하여 구현하였다. 그 결과 19,000개 이상의 다수 문서 집합에서 73%의 정확율로 보다 정확한 검색이 가능하게 되었다.

키워드 : 인접 관계, n-gram, 용어빈도-역문헌빈도, 순위 기반 검색, 정보 검색

Abstract

Information retrieval is a method to search the needed data by users. Generally, when a user searches some data in the large scale data set like the internet, ranking-based search is widely used because it is not easy to find the exactly needed data at once. In this paper, we propose a novel ranking-based search method based on sequence and adjacency relationship of user query by the help of TF-IDF and n-gram. As a result, it was possible to find the needed data more accurately with 73% accuracy in more than 19,000 data set.

Key Words : adjacency relationship, n-gram, TF-IDF, ranking-based search, infromation retrieval

1. 서 론

정보 검색(Information retrieval)은 대량의 자료에서 사용자의 요구를 충족시키는 정보를 찾아내는 방법이며, 일반적으로 데이터베이스가 대규모의 자료 집합을 관리하기 위해서 사용된다. 또한 사용자가 찾는 정보는 주로 비구조적인 속성의 문서이기 때문에 대규모의 문서들 사이에서 사용자가 원하는 적절한 문서를 찾아낸다는 것은 쉬운 일이 아니다. 따라서 다수의 문서들 중에서 정보를 찾아낼 때는 문서에 순위를 부여하여 정렬 후 사용자에게 순위에 따라 문서들을 정렬하여 제시하는 방법을 일반적으로 사용한다[1].

이를 위해 사용되는 일반적인 방법들로서 가중치 구역 점수 계산(Ranked boolean retreival)[1], 페이지랭크(Page-Rank)[2]와 용어빈도-역문헌빈도(Term Frequency - Inverse Document Frequency, TF-IDF)[3]가 있다.

먼저 가중치 구역 점수 계산은 문서의 각 구역, 즉 제목, 본문, 저자와 같은 부분에 대하여 각각 따로 가중치를 정해놓고 사용자 질의가 이 구역들과 일치하는 부분이 있을 경우 사용자 검색어가 나타난 구역에 대해서만 가중치를 이용한 점수를 부여하는 것이다. 하지만 이 방식은 각 구역에 대해서 가중치의 값을 어떻게 결정할지를 판단할 필요가 존재한다.

다음으로 페이지 랭크는 인터넷 상에 존재하는 다수 페이지들의 링크관계를 파악하여 어느 특정 페이지를 다수의 다른 페이지들이 링크하고 있다면 그 페이지가 다른 페이지보다 대중성이 높다고 파악하고 이에 대해 높은 가중치를 부여한다. 하지만 대중성이 높다고 해서 그게 반드시 사용자의 요구와 부합하는 페이지라고 보기에는 어렵다는 문제가 존재한다.

용어빈도-역문헌빈도는 단어와 문서 혹은 단어와 단어간의 관계를 나타낸다. 먼저 용어 빈도는 문서에 단어가 포함되는 수만큼 가중치를 부여한다는 것이다. 하지만 이는 식별력이 없는 단어, 즉 불용어(Stop words)에도 가중치를 부여할 수가 있다.

역문헌빈도는 이러한 점을 보완한다. 역문헌빈도는 문헌의 집합에서 특정 용어를 포함하는 문서의 수가 많을수록 그 용어에 대한 가중치를 감소시키는 방법이다. 이

접수일자 : 2011년 11월 19일
완료일자 : 2011년 12월 16일
* 본 논문은 본 학회 2011년도 추계학술대회에서 선정된 우수논문입니다.
* 본 연구는 2011년 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 기초사업, 과제번호: 2011-0027095에 의해 수행되었습니다. 연구비 지원에 감사드립니다.
+ 교신저자

렇게 용어빈도와 역문헌빈도를 조합하는 방식으로 사용자의 질의에 대한 문서의 순위를 설정하여 검색 결과를 낼 수가 있다.

하지만 위와 같은 용어빈도-역문헌빈도는 단어 간의 순서와 인접성을 고려하지 않는다는 단점을 가지고 있다. 단어와 문서 사이의 빈도만을 고려하기 때문에 사용자 질의에 있는 단어들은 포함하나 사용자가 요구하는 내용과는 전혀 다른 문서가 검색될 수 있기 때문이다.

본 논문에서는 용어빈도-역문헌빈도에 더해 사용자 질의 문장에 n-gram 기법을 응용한 단어의 인접성 및 순서에 대한 가중치를 추가한 새로운 검색 기법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구에 대해서 기술하고 있다. 그리고 3장에서는 본 논문에서 제안하는 검색 기법에 대해서 자세히 기술하고 있으며 4장과 5장에서는 실험 및 결론에 대해서 논하고 있다.

2. 관련 연구

논문의 본 장에서는 문서 집합에서 순위 기반 검색을 수행하기 위해서 사용되고 있는 방법들에 대해서 알아보도록 하겠다.

2.1 가중치 구역 점수 계산

가중치 구역 점수 계산은 문서의 구역별로 다른 가중치를 두어 사용자 질의에 대해 순위 점수를 계산하는 방식이며 아래의 식(1)과 같이 정의된다.[1]

$$\sum_{i=1}^{l} g_i s_i = S_a \tag{1}$$

여기에서 l은 문서 구역의 수, g_i는 구역별 가중치, s_i는 0또는 1로 정의되며 이는 구역 g_i에서 사용자 질의와의 일치 여부에 따른 불리언 점수이다. 이 s_i를 각각의 가중치 g_i에 곱한 후 다 더한 것이 최종 점수 S_a이다.

2.2 페이지랭크

페이지랭크는 웹 그래프에서 페이지간의 링크관계로 그 값이 설정되며, 타 페이지에서 많이 링크가 되어 있을수록 높은 순위 점수를 가진다. 이는 아래의 식(2)와 같이 나타내어진다.[4]

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)}\right) \tag{2}$$

여기에서 A는 점수 계산 대상의 페이지이며 $T_1 \ldots T_n$은 A로 연결되어 있는 페이지들이다. d는 0부터 1사이로 설정되는 감쇠지수이며 일반적으로 0.85로 설정된다. 그리고 C는 페이지 T_i에서 페이지 밖으로 나가는 링크의 수를 나타낸다.

2.3 용어빈도-역문헌빈도

용어빈도-역문헌빈도는 단어와 문서간의 관계에 따른 가중치를 결정하기 위해 사용되는 기법이며, 아래의 식(3)과 같이 정의된다.

$$tf\_idf(t,d) = tf(t,d) \times idf(t)$$
$$idf(t) = \log\frac{N}{df(t)} \tag{3}$$

식(1)에서 t는 용어, d는 문서, tf는 용어빈도, df는 t를 포함하는 문서들의 수, N은 집합 내 문서들의 총 수, idf는 역문헌빈도, 마지막으로 tf_idf는 용어빈도-역문헌빈도를 나타낸다. 이 용어빈도-역문헌빈도는 사용자 질의 및 문헌에 적용되어 각각의 벡터로 표현되며, 벡터 사이의 내적을 통한 벡터 유사도를 계산함으로 써 순위 점수를 얻는 것이 가능하다[1].

또한 이 외에도 순위 기반 검색에 대해서 다양한 연구가 이루어지고 있다. 먼저 블로그 포스트 페이지에 대한 랭킹을 위해 몇몇의 연구가 있어왔다.[5][6] “블로그-랭크” 알고리즘[6]은 높은 트랙백 연결성과 댓글을 통한 사용자의 반응성을 가진 페이지를 좋은 페이지로 나누고, 특정 주제에 높은 추천수를 가진 블로거가 작성한 페이지를 더 높이 평가한다. 이 알고리즘은 특정주제에 대한 블로그의 명성과 페이지의 트랙백 연결성, 페이지의 사용자 반응성에 대한 평가 모델을 정의 한다. 정의된 평가 모델은 특정 주제에 대해 블로거들이 작성한 포스트들에 대한 평균 점수와 특정 주제에 대한 블로거의 활동 점수를 적용하여 계산한다.

또한 XML문서 순위화 기법에 대한 연구도 존재한다.[7] 용어적, 구조적으로 다양하게 존재하는 환경이라도 동일한 용어 및 구조를 사용한 환경과 마찬가지로 최상위 순위 정보 검색방법을 제공하며 이 기법은 XML문서가 가지는 용어적, 구조적 다양성을 고려하여 사용자 질의를 보다 확장된 개념으로 처리해 준다.

그리고 사용자 프로파일을 이용하여 문서 순위를 결정하는 방법도 연구되었다.[8] 이 방법은 프로파일을 이용하여 사용자의 선호도를 나타내고 문서들의 잠재적 구조를 사용자의 선호도와 비교하여 사용자 프로파일과 문서 사이의 유사성을 비교하게 되며 적합 정도에 따라 사용자에게 최적의 문서를 제공한다.

다음으로 기존 웹 그래프를 이용한 보다 효과 적인 페이지 순위 산정을 위한 연구도 있다[9] 웹 문서의 순위 평가 알고리즘인 HITS 알고리즘의 문제인 문서 내의 링크 빈도수만을 고려하고 입력값인 웹 문서 집합의 특성에 의존적이라는 것을 개선하기 위하여 링크 기반 웹 검색 엔진들로부터 얻어진 문서 집합으로 초기 집합을 생성 및 향상된 HITS 알고리즘을 사용하였다.

3. 제안된 기법 및 구현

3.1 사용자 검색 질의 단어의 순서 및 단어 간의 인접 관계에 기반한 검색 기법

n-gram은 기호의 연쇄를 나타낸다. 일반적으로 인간의 자연어가 정규문법을 따르지 않기 때문에 문장 또는 음성의 인식을 확률적으로 처리하기 위해 사용되며 대표적인 응용으로 언어처리에 사용되는 n-gram을 이용한 마르코프 모델인 n-gram 모델이 존재한다[10]. 본 논문에서는 이를 응용한 n-gram 문장이라는 개념을 사용한다. 그림 1은 ‘I have a question’ 이라는 n-gram 문장에서 나올 수 있는 단어들의 연쇄에 대한 예시를 나타낸다.

n-gram 문장 : 'I have a question'
2-gram 연쇄 문장 : 'I have', 'have a', 'a question'
3-gram 연새 문장 : 'I have a', 'have a question'
4-gram 연새 문장 : 'I have a question'

그림 1. n-gram 문장 예시
Fig 1. Example of n-gram sentence

그림 1과 같이 예시로 제시된 'I have a question' 이라는 문장은 n-gram에서 n의 값만큼의 단어를 가지는 부분 문장으로 나누어질 수 있다. 이러한 n-gram 문장과 거기서 파생되는 n-gram 연쇄 문장들은 단어 간의 순서 및 인접도의 계산에 사용된다. 이렇게 문장을 분해하는 이유는 전체 문장뿐만이 아닌 부분 문장에 대해서도 단어 간의 순서 및 인접도를 고려하기 위해서이다. 이제 논문에서 제시하는 문서 순위 결정을 위한 식은 아래의 식(4)와 같다.

$$W=\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{daf({}_nq_i,Q_n)}{1+\log(dis(q_i,m_{ij}))}+1$$
$$Q_q=\{q_i\mid q_i\in Q,1\le i\le n\} \quad (4)$$
$$M=\{m_{ij}\mid m_{ij}\in d,1\le i\le n\}$$

여기에서 Q_q는 질의 Q의 n-gram 연쇄 문장인 q_i의 집합, n은 질의 Q의 n-gram 연쇄 문장의 수, M은 문서 d에서 각 q_i와 비교해 문서의 단어 출현 순서에 따라 조합이 일치하는 부분인 m_{ij}의 집합이며, k는 각 q_i에 대해 m_{ij}가 일치하는 개수를 가리킨다. 또한 ${}_nq_i$는 각 q_i문장의 단어 수, Q_n은 질의 Q의 단어 수, dis는 Q_q와 문서 d사이의 비유사도를 측정하기 위한 함수이며 이는 단어 사이의 간격에 비례한다. 마지막으로 daf함수는 감쇠율(damping factor)을 계산하기 위한 함수이다.

예를 들어, 문서가 d={my life my this is my new life} 의 7단어로 이루어져있고, 사용자가 질의로 Q={in my life}를 입력했다고 하자. 이때, Q_q는 질의의 n-gram 연쇄 문장의 집합이므로 Q_q={q_1=in my life, q_2=in my, q_3=my life} 로 정의된다. 문서 내에서 단어가 출연 하는 순서에 따라 각 q_i가 문서 d와 일치하는 부분을 정의한다면 M={m_{31}='my life'[1:2], m_{32}='my new life'[6:8]} 가 된다. 여기서 괄호'[ ]'안의 숫자는 문서 d에서 m_{ij} 가 시작되는 위치인 s와 끝이 나는 위치인 e를 [s:e] 형태로 가리키며, q_1, q_2와 일치하는 부분은 없으므로 m_{1j}, m_{2j}는 M의 원소로 되지 않는다. m_{32}가 q_3와 단어가 완전히 일치하지 않으면서도 M의 원소가 된 이유는 q_3와 단어가 나오는 순서가 일치하기 때문이다. 물론 이러한 경우에는 가운데 'new'라는 단어가 있으므로 q_3와는 완전히 일치하지 않으며 질의와 일치하는 단어가 서로 떨어져 있는 만큼 비유사도가 올라가게 된다.

비유사도를 측정하기 위한 함수인 dis는 m_{ij}에서 q_{ij}와 일치하는 단어 사이의 간격으로 이루어지며, 식(5)와 같이 정의된다.

$$dis(q_i,m_{ij})=\begin{cases}\text{if } q_i\text{와 } m_{ij}\text{의 단어 출연 순서 일치시}\\ \quad \{m_{ij}\}-\{q_i\}\text{집합의 원소의 개수}\\ \text{if } q_i\text{와 일치하는 } m_{ij}\text{없을 시}\\ \quad \infty\end{cases} \quad (5)$$

식(5)에서 $\{m_{ij}\}$와 $\{q_i\}$는 각각 m_{ij}와 q_i를 구성하는 단어들의 집합이며 dis함수는 이 두 집합의 차집합의 원소의 개수이다. 예를 들어, q_3과 m_{31} 은 서로 간에 단어의 나열이 정확히 일치하므로 $dis(q_3,m_{31})$=0 이다. 하지만, $dis(q_3,m_{32})$=1 이 된다. 이는 'new' 라는 단어가 'my'와 'life'의 사이에 놓여 있으며 이로 인해 'my'와 'life'의 간격이 1이기 때문이다. 하지만 만약 q_i에 부합되는 m_{ij}가 없을 경우에는 유사한 점을 찾을 수가 없으므로 비유사도는 무한값인 ∞로 설정되며, 이는 위의 예시 중 q_1, q_2에 해당된다.

단어 간의 상대거리 차이를 상대적으로 보존하면서 또한 완화시키기 위해 비유사도의 정규화에는 log함수를 사용한다. 그리고 daf함수의 결과를 이 정규화 값으로 나누어 줌으로서 각 비유사도를 유사도로 변환한다. daf함수의 정의는 아래의 식(6)과 같다.

$$daf({}_nq_i,Q_n)=(\frac{{}_nq_i}{Q_n})^c \quad (6)$$

daf함수의 정의로서 $({}_nq_i/Q_n)^c$가 사용되는 이유는 n-gram 연쇄 문장의 단어 수를 유사도에 고려하여 감쇠율을 조정할 필요가 있기 때문이다. 다시 말해 2-gram 집합의 각각의 연쇄 문장들이 문서와 일치하는 부분에 대해 점수를 주는 것 보다는 3-gram 또는 4-gram 문장이 문서와 일치하는 부분에 더 높은 가중치를 주어야 할 것이기 때문이다.

이를 위해 각 n-gram 연쇄 문장의 단어 수인 ${}_nq_i$가 작아질수록 $({}_nq_i/Q_n)^c$ 의 값 역시 작아지며 이는 질의 단어의 부분으로 이루어진 문장보다는 질의 단어의 전체로 이루어진 문장이 유사도에 더 많은 영향을 미치는 것을 의미한다. 또한 각 n-gram 단계 사이에서의 가중치 편차를 조절하기 위해 상수 c를 적당한 값으로 조절할 필요가 있으며, 본 논문에서는 이 값으로 2를 사용하고 있다.

마지막으로 질의와 문서간의 최종 유사도 점수는 위의 2장 관련연구에서 논한 tf_idf를 이용한 벡터 유사도 점수와 지금까지 논한 단어 간의 순서 및 인접도에 대한 유사도 점수를 이용하게 된다. 문서 순위를 정하기 위한 최종적인 점수인 S는 아래의 식(7)과 같다.

$$S(Q,d)=(Q_{tf\times idf}\cdot d_{tf})\times W \quad (7)$$

여기서, $Q_{tf\times idf}$ 는 사용자 질의 Q의 용어빈도-역문헌빈도 정규화 벡터, d_{tf}는 문서의 용어빈도 벡터를 의미한다. 일반적으로 질의에는 tf와 idf를 모두 적용하나 문서에는 유효성과 능률의 이유로 이를 적용하지 않는다고 알려져 있다[1] $Q_{tf\times idf}$와 d_{tf} 의 내적은 벡터유사도를 통한 순위 점수를 의미하며, 여기에 단어 간의 순서 및 인접도에 대한 가중치 W를 곱한 것이 최종 점수 S가 된다.

3.2 제안된 기법의 구현

문서의 빠른 검색을 위해서는 문서의 정보를 일정한 규칙을 통해 저장하고 있는 인덱스를 생성하는 것은 필수적인 요소이다. 먼저 인덱싱은 키워드를 중심으로 해서 문서를 연결하는 역-인덱싱 기법을 사용하며 사용되는 문서는 SGML언어로 저장되어 있는 로이터 문서 집합이다. 로이터 문서 집합 중 date(날짜), topics(주제), place(장소), title(제목), body(본문) 정보를 가지고 있는

문서들에 한정해 이들 문서를 유효하다고 판단하고 인덱싱이 수행이 된다. 그림 2는 본 논문에서 제안하는 기법에서 실제로 구현되어 있는 인덱싱 과정을 나타낸다.

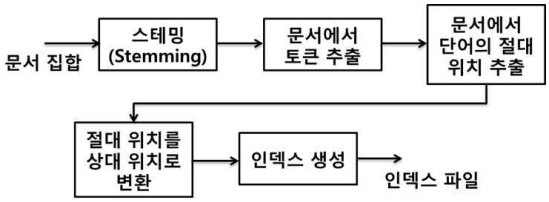

그림 2. 인덱스 생성 과정
Fig 2. Process of index creation

유도된 단어나 변화된 단어를 제거하고 단어의 원형을 추출하기 위한 스테밍 과정은 인덱스의 단어 수가 기하급수적으로 늘어나는 것을 방지하는 역할을 하기 때문에 이는 인덱스 생성과정에서 필수적인 전처리 과정이다. 본 논문에서는 이 스테밍 과정을 수행하기 위하여 포터 스테머(porter's stemmer)[11]를 사용하였다. 이 과정을 통해서 추출된 단어 중 대문자들은 모두 소문자로 변환이 되어 인덱스에 저장이 되게 된다.

본 논문에서 제안하는 기법을 위해 인덱싱 과정에서 추가적으로 수행되는 중요한 점 중 하나는 문서에서 단어의 위치를 추출한다는 점이다. 본 논문에서 제안하는 기법은 단어와 단어간의 인접관계를 필수적으로 사용하므로 단어간의 위치관계는 필수적이다. 이를 위해 인덱싱과정에서는 먼저 토큰 추출된 단어들의 절대 위치를 찾고 그 다음 절대 위치를 단어간의 상대 위치로 바꾸어 준다.

예를 들어, 'A & B'에서 기호 '&'은 인덱싱을 할 때에는 제거되는 의미 없는 문자라고 하자. 이때, A와 B의 절대 위치는 각각 1, 3이므로 절대 위치로는 서로 간에 1의 간격이 있다고 볼 수 있다. 하지만 의미 없는 기호 '&'을 인덱싱 과정에서 제거한 후 단어의 위치를 상대위치로 따지면 A와 B의 위치는 각각 1, 2가 되므로 이 두 글자가 서로 인접해 있다는 것을 알 수가 있다. 따라서 인덱스에서 단어들의 위치는 모두 추출된 단어 사이의 상대위치로서 고려한다.

다음으로는 사용자 질의 처리 부분이다. 사용자 질의로는 자연어를 사용한다. 입력으로 받은 자연어는 전처리 과정을 통해서 내부에서 사용할 수 있는 형태로 변형이 되며, 이 전처리된 질의와 동시에 사전에 생성해 놓았던 인덱스를 이용해서 검색을 수행하게 된다. 자연어 질의의 정규화 과정은 위에서 설명했던 문서를 인덱싱하는 과정과 유사하다. 다음의 그림3은 질의어가 전처리된 후의 결과를 보여준다.

질의어 : The project extends through 1991
질의 전처리 결과
The → the, tf_idf값 : 0.027
project → project, tf_idf값 : 0.435
extends → extend, tf_idf값 : 0.494
through → through, tf_idf값 : 0.335
1991 → 1991, tf_idf값 : 0.672

그림 3. 질의어 전처리 예시
Fig 3. Example of preprocessing a user query

사용자 질의역시 인덱싱 과정 처럼 포터 스테머를 이용해 어근을 추출하고 소문자로 변환한다. 위의 그림 3에서 이 결과로 The는 the로 extends는 extend로 변환된 것을 알 수가 있다. 다음으로 추출한 각 단어에 대해서 *tf_idf* 값을 계산한다. 그림 3을 보면 다른 단어에 비해 the의 *tf_idf*값이 낮은 것을 알 수 있는데 이는 the가 다른 문서에서도 매우 많이 나오는 불용어이기 때문에 이를 감안해 *tf_idf*값이 낮아졌기 때문이며 전처리 결과만 보면 1991이라는 단어가 문서 집합에서 가장 출연하지 않은 단어라는 것을 알 수가 있다. 이 전처리 결과와 인덱스를 이용해 *tf_idf*를 이용한 벡터유사도와 본 논문에서 제안하는 유사도 검출 기법을 사용하여 입력된 사용자의 질의어와 문서집합의 각 문서 간의 유사도 점수를 계산하게 된다.

최종적으로 구현된 프로그램은 웹서버로 Apache Tomcat 6.0을 사용하였고 JAVA와 JSP를 이용하여 웹 기반으로 만들어졌으며 구현을 위한 컴퓨터는 사양이 Intel(R) Core(TM) i5 CPU에 RAM이 4GB인 것을 사용하였다. 아래의 그림 4는 'telephone from paris'라는 질의어에 대한 실제 구현된 프로그램의 검색 결과를 보여주며 검색결과에서 'score'항목이 문서와 질의어 간의 유사도 점수를 나타내고 있다.

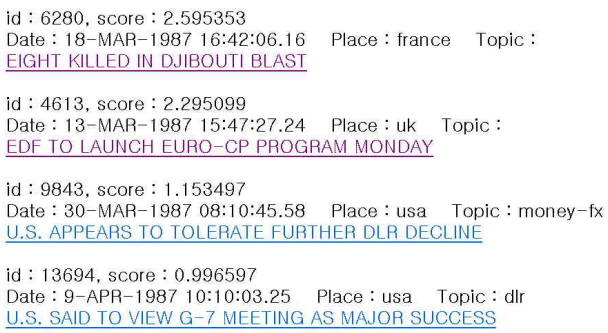

그림 4. 구현된 프로그램의 실제 검색 결과
Fig 4. Search result of the implemented program

그림4를 보면 'telephone from paris' 질의어에 대해 다른 문서들보다 높은 점수를 가지는 문서가 'EIGHT KILLED IN DJIBOUTI BLAST'와 'EDF TO LAUNCH EURO-CP PROGRAM MONDAY'의 두 가지가 있는데 이들 문서를 보면 가장 높은 점수가 나온 첫 번째 문서에는 'telephone from paris' 라는 문장이 있고 두 번째 문서에는 'telephone call from paris' 라는 문장이 존재한다. 이를 보면 문장이 완전히 일치하는 부분이 존재할 시에는 가장 높은 유사도 점수를 그 문서에 부여하며 완전히 일치하는 부분이 없더라도 단어의 순서가 일치하고 또한 단어 간의 인접도가 가깝다면 역시 높은 점수를 부여한다는 것을 알 수가 있다.

4. 실험 및 분석

실험은 주어진 키워드에 대하여 평균정확율(Mean Average Precision, MAP)을 측정하는 것으로 이루어졌다. 실험에 사용된 키워드 집합 및 문서 집합은 아래의 표 1과 같다.

표 1. 실험 수행 환경.
Table 1. The experimental environment

구 분	세 부 내 용
문헌 집합	· Reuters-21578 문헌집합 사용 (총 19,043개 문서)
검색어 집합A	· 2004-2005년도 영국 구글 뉴스/시사 부분의 인기 검색어 중 총 50개를 선정
검색어 집합B	· Reuters-21578 집합에서 임의로 문서 및 단어를 선택하여 검색어를 총 50개를 선정

문서 집합으로는 실험용 문서 집합으로서 잘 알려진 Reuters-21578 문서 집합을 사용하였으며 이 중 적합하다고 판단되는 19,043개의 문서를 골라내서 실험을 수행하였다. 그리고 검색어의 집합은 2개 집합으로 나누었으며 하나는 웹상에서 사용되는 검색어를 그대로 사용하였고 다른 하나는 문헌집합에서 임의로 검색어를 50개를 선정하여 검색어 집합을 생성하였다. 이 두 검색어 집합에 대해서 실험을 수행한 결과는 표 2와 같다.

표 2. 실험 결과
Table 2. The experimental results

사용 방법	검색어 집합	세부 내용 (상위 5개의 문서를 사용)
용어빈도-역문헌빈도	검색어 집합A	평균 정확율 56%
제안하는 방법	검색어 집합A	평균 정확율 61%
	검색어 집합B	평균 정확율 73%

실험 결과, 먼저 검색어집합A에 대해 용어빈도-역문헌빈도 사용시의 평균 정확율은 56%의 결과가 나왔으며 본 논문에서 제안하는 방법으로는 61%의 결과가 나왔다. 또한 검색어집합B도 제안하는 방법으로 73%의 평균 정확도 결과가 나왔다. 이 결과는 제안하는 방법이 용어빈도-역문헌빈도보다 효과적으로 검색을 수행한다는 것을 보여준다. 그리고, 제안하는 방법의 검색어 집합A에 대한 평균 정확율은 높은 수치라고 볼 수는 없으나 이러한 수치가 나오는 이유를 Reuters-21578 문서 집합이 검색어집합A와 시기상으로 그 주제가 맞지 않는 부분이 많을 수 있다는 것을 그 이유로서 생각해 볼 수가 있다.

실제로 Reuters-21578에서 임의로 검색어를 선택하였으므로 시기상으로 그 주제가 Reuters-21578문서 집합과 유사할 수 있는 검색어집합B의 제안하는 방법에 대한 평균정확율은 검색어집합A 보다 약 12%가 상승된 73%라는 수치를 보였다. 이는 문서 집합이 인터넷과 같이 지금보다 훨씬 넓은 주제를 포괄할 수 있게 된다면 그에 비례해 평균정확율도 상승할 수 있다는 얘기가 된다.

실험 결과, 제안하는 방법은 용어빈도-역문헌빈도만을 사용했을 때보다 더 높은 평균 정확률을 보여줬으며 또한 검색어집합B를 이용한 실험을 통해 문서 집합이 더 많은 내용을 포괄할 수 있다면 평균정확도가 더 높아질 수 있다는 것을 알 수 있었다.

5. 결 론

정보 검색은 대량의 자료에서 사용자의 요구를 충족시키는 정보를 찾아내는 방법이며, 일반적으로 대규모 자료 집합이 검색 대상이 된다. 따라서 사용자가 한 번에 자신이 찾는 문서를 대규모 문서 집합에서 찾아낸다는 것은 쉽지 않은 일이며 이를 극복하기 위해 일반적으로 사용자 질의와 문서간의 유사도를 측정하여 유사도가 높은 것들만을 사용자에게 제시해주는 순위 기반 검색 방법이 주로 사용된다.

본 논문에서는 기존의 용어빈도-역문헌빈도에 더해서 n-gram연쇄 문장을 이용한 단어 순서 및 인접성 가중치를 추가한 새로운 검색기법을 제안하였다. 이는 기존의 *tf_idf*벡터 유사도 방식이 사용자 질의의 단어 순서와 인접 관계를 고려하지 않는 다는 점을 보완하기 위함이다.

제안한 방법에 대한 실험에는 웹에서 50개의 검색어를 모아온 검색어 집합A와 실험에 사용된 문서 집합인 Reuters-21578에서 임의로 선택된 검색어 집합B가 사용되었으며 검색어 집합A의 용어빈도-역문헌빈도 평균정확율은 56%, 제안하는 방법에 대한 평균 정확율은 61%가 나왔고 검색어 집합B의 제안하는 방법에 대한 평균 정확율은 73%를 보였다. 제안하는 방법은 용어빈도-역문헌빈도만을 사용했을 때보다 더 높은 평균 정확률을 보여줬으며 또한 검색어집합B는 문서 집합의 범위에 따라 평균정확도가 더 높아질 수 있다는 것을 보여준다.

참 고 문 헌

[1] C. D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[2] S. Brin and L page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine, Proc. of 7th international conference on World Wide Web,” pp. 107-117, 1998 .

[3] K. S. Jones, “IDF term weighting and IR research lessons,” Journal of Documentation, Vol. 28, pp.11-21, 1972.

[4] S. Brin, “The Anatomy of a Large Scale Hypertextual Web Search Engine,” International world wide web conference, pp. 107-118, 1998

[5] 김정훈, 윤태복, 이지형, “효율적인 블로그 검색을 위한 블로그-랭크 알고리즘,” 한국정보과학회 2008 가을 학술발표논문집, Vol. 35, No. 2, 2008

[6] 김정훈, 윤태복, 이지형, “블로그의 구조적 특성을 고려한 효율적인 블로그 검색 알고리즘,” 정보과학회논문지. 소프트웨어 및 응용, Vol. 36, No. 7, 2009

[7] 김현주, 박소미, 박석, “확장된 질의 처리를 위해 경로간 의미적 유사도를 고려한 XML문서 순위화 기법,” 정보과학회논문지. Journal of KIISE. 데이터베이스, Vol. 37, No. 2, pp.113-120, 2010

[8] 김용호, 김형균, 최광미, “사용자 프로파일을 이용한 문서순위 결정 방법,” 한국해양정보통신학회 2005년도 추계종합학술대회, Vol. 9, No. 2, pp.615-618, 2005

[9] 김분희, 한상용, 김영찬, “웹 문서 중요도 평가를 위한 적합도 향상 HITS 알고리즘 설계,” 한국전자거래학회지, Vol. 8, No. 2, pp.23-31, 2003

[10] John Coleman, Introducing Speech and Language Processing, Cambridge University Press, 2005

[11] Martin Porter. 2001. The Porter Stemming Algorit hm. http://www.tartarus.org/martin/PorterStemmer/ind ex.html

저 자 소 개

소병철(Byung-Chul So)

2010년 : 동국대학교 컴퓨터공학과(공학사)
2010년 ~ 현재 : 동국대학교 컴퓨터공학과 석사과정

관심분야 : 모바일 로봇, 정보검색, 인간-로봇 상호작용
E-mail : sbc10620@naver.com

정진우(Jin-Woo Jung)

1997년 : 한국과학기술원 전기 및 전자공학과(공학사)
1999년 : 한국과학기술원 전기 및 전자 공학과(공학석사)
2001년 ~ 2002년 : 일본 동경대학교 기계정보공학과 대학원 방문연구원
2004년 : 한국과학기술원 인간친화 복지 로봇시스템 연구센터 박사후연구원
2006년 ~ 현재 : 동국대학교 컴퓨터공학과 조교수

관심분야 : 인간-로봇 상호작용, 다개체 협력로봇, 소프트컴퓨팅, 생체측정, 지능로봇
E-mail : jwjung@dongguk.edu