용어 가중치 재 산정을 이용한 검색 시스템

황선욱[O] 김혜정 손기준 이상조
경북대학교 언어 · 정보 연구실
{hodduk[0] hjkim kjson}@sejong.knu.ac.kr, sjlee@knu.ac.kr

Retrieval System Using Term Reweighting

Sun-Wook Hwang[0] Hae-Jung Kim Ki-Jun Son Sang-Jo Lee
Department of Computer Engineering Kyungpook National University Daegu, Korea

요 약

색인 파일은 정보 검색 시스템에서 중요한 구성 요소 중에 하나이다. 스테밍을 하여 색인 파일을 구성하게 되면 파일의 크기를 줄일 수 있고 재현율을 높이는데 효과적이다. 하지만, 과도한 스테밍으로 구성이 된 색인 파일은 원형에 대한 데이터 손실을 가지고 오기 때문에 너무 많은 문서가 검색되어 사용자가 문서를 찾는데 많은 시간이 소요되고 정확률도 떨어진다.

본 논문에서는 정보 검색 시스템에서 검색의 효율성을 높이기 위해 사용하는 색인 파일을 스테밍 한 것과 스테밍 하지 않은 파일로 구성하였다. 스테밍 한 색인 파일은 질의어와 문서 사이의 유사도를 계산하기 위하여 이용되며, 스테밍 하지 않은 파일은 스테밍 했을 때 검색된 문서들 중에서 데이터 손실로 인한 잘 못된 문서 순서를 재조정 해 주기 위하여 이용된다. 본 논문에서는 높은 검색 효과를 제공하는 기존의 벡터 공간 모델을 검색 성능 평가 척도 중의 하나인 R-정확률을 이용하여 비교 평가하였다. 본 논문에서 제안하는 시스템이 문서 상위 100위까지에 대하여 일반 벡터 모델 보다 최고 21%의 좋은 성능을 보였다.

1. 서 론

사용자가 원하는 정보 탐색 시간을 최소화하기 위한 방법으로 정보 검색 시스템에 대한 연구가 증대되고 있다. 정보 검색 시스템의 중요한 목적 중에 하나는 사용자의 요구를 나타내는 질의어와 문서 사이의 유사도를 계산하여 사용자가 원하는 관련 있는 문서를 상위에 위치하게 하는 것이다. 사용자는 상위에 위치한 문서부터 검토를 하며 원하는 문서를 찾는데 소요되는 시간을 최소화 할 수 있다.

색인 파일은 정보 검색 시스템 구성 요소 중의 하나이며 색인 파일 구성 방법에 따라 정보 검색 시스템의 성능을 좌우한다. 스테밍하여 색인 파일을 구성하게 되면 색인 파일의 크기를 줄이고 검색의 효율을 높이는데 효과적이다. 하지만, 과도한 스테밍으로 인한 데이터 손실은 검색하였을 때 너무 많은 문서가 검색되어 사용자가 문서를 찾는데 많은 시간이 소요되고 정확률도 떨어진다.

본 논문에서는 벡터 공간 모델에서의 용어 가중치를 재 산정하여 기존의 벡터 공간 모델과 R-정확률을 이용하여 비교 평가하였다. 문서 상위 100까지에 대하여 일반 벡터 모델 보다 최고 21%의 좋은 성능을 보였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 문서 순위화와 관련 있는 연구들에 대해 알아보고 과도한 스테밍의 문제점을 지적한다. 3장에서는 문서 순위화를 위한 검색 시스템을 소개한다. 4장에서는 R-정확률을 사용하여 높은 검색 효과를 제공하는 벡터 모델과 비교 평가한다. 마지막으로 5장에서는 결론 및 향후과제를 제시한다.

2. 관련연구

문서를 순위화 하는 방법에는 사용자 위주로 문서를 순위화하는 방법, 시소러스를 기반으로 불린 모델을 이용한 문서 순위화 하는 방법, 벡터 공간 모델을 이용한 문서 순위화하는 방법 등이 있다.

사용자 위주로 문서를 순위화 하는 방법은 사용자의 질의어와 검색된 문서들 사이의 유사도에 따라 문서의 순위는 결정되고 사용자는 결정된 문서를 참고하는 방법이다[1,2,3].

시소러스를 기반으로 불린 모델을 이용한 문서를 순위화하는 방법은 색인어들 사이의 연관 관계를 이용함으로서 문서 값을 계산한다[3]. 색인어들 사이의 연관 관계를 이용하면 높은 검색 효율을 제공하지만 가중치 연산에 대한 효율적인 연산 방법을 지원하지 않는다.

벡터 공간 모델을 이용한 문서순위화 방법은 질의어와 문서를 벡터 공간으로 표현하여 나타내며, 질의어와 문서 사이의 유사도 값을 계산하여 관련 있는 문서를 나타내는데 효과적이다[4,5]. 하지만, 과도한 스테밍으로 인한 데이터 손실 때문에 문서 순위가 잘 못 되는 경우가 발생한다. 예를 들면, 질의어가 arms일 경우, 검색된 문서 순서가 첫 번째 문서는 arm이고 두 번째 문서가 arms일 경우 스테밍을 하게 되면 둘 다 arm이 되기 때문에 문서 순서에는 변화가 없게 된다.

3. 용어 가중치 재 산정을 이용한 검색 시스템

본 논문에서 제안한 검색 시스템은 전처리 모듈과 검색 모듈로 구성된다. 그림 1은 전체 시스템의 흐름도이다.

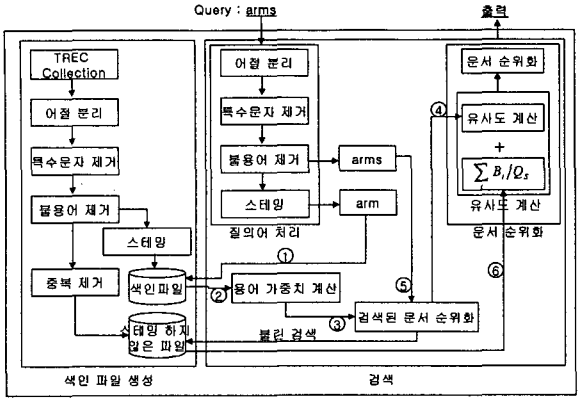

그림 1 검색 시스템의 흐름도

전처리 과정에서는 문서 집합에서 어절을 분리 및 특수 문자를 제거하고 불용어 리스트를 사용하여 불용어를 제거한다. 색인 파일은 스테밍 작업을 수행하여 생성하고 원시 파일은 용어의 중복을 제거하여 생성하게 된다. 이때 사용한 문서 집합은 TREC 테스트 컬렉션에 들어 있는 WSJ(Wall Street Journal), AP(Associated Press) 기사를 대상으로 하였다.

다음 단계에서는 질의어를 arms라고 입력하였을 때의 검색되는 과정을 보여주고 있다. 먼저, 질의어와 관련 있는 문서를 검색하기 위하여 질의어 처리 모듈에서 스테밍을 하여 색인 파일을 검색하는 부분이 처리 ①이 된다. 처리 ②는 질의어와 관련 있는 모든 문서를 검색하여 각 문서에 대한 용어 가중치를 계산하게 된다. 용어 가중치는 Salton[6,7]이 제안한 TF*IDF 방법으로 용어의 가중치를 계산한다. 용어의 가중치 계산 방법은 식 (1)과 같다.

$$W_{i,j} = tf_{i,j} * \log(N/d_j) \qquad (1)$$

tf_{ij} : 문서 i에서 용어 j의 빈도수

$\log(N/d_j)$: 전체 문서 N에서 용어 j가 나타난 문서의 역 문헌 빈도수

처리 ③은 용어 가중치에 대한 관련 문서들을 순위화한다. 처리 ④는 질의어와 문서 사이의 유사도 값을 계산한다. 처리 ⑤는 질의어를 스테밍 했을 때 검색된 문서들 중에서 데이터 손실로 인한 잘 못된 문서 순서를 재조정 해주기 위하여 스테밍 하지 않은 용어를 검색한다. 처리 ⑥은 스테밍 하지 않은 질의어를 불린 검색하여 일치한 용어들에 대하여 전체 질의어 수를 나누어 값을 정량화 한다.

다음 단계로는 처리 ④와 처리 ⑥으로 계산된 값을 이용하여 질의어와 문서 사이의 유사도를 계산한다. 유사도 계산은 코사인 유사도 계산 방법을 이용하며 계산 방법은 식 (2)와 같다.

$$SIM(Q, D_d) = \frac{Q_f \bullet D_d}{|Q_f| \times |D_d|} + \sum_t B_t / Q_s$$

$$= \frac{\sum_t (\log N/df_t) \times tf_t \times (\log N/df_t)}{\sqrt{\sum_t (\log N/df_t)^2} \times \sqrt{\sum_t \{tf_t \times (\log N/df_t)\}^2}} + \sum_t B_t / Q_s \qquad (2)$$

$Q_f \bullet D_d$: 처리 ①에서 전체 질의어와 특정 문서 d와의 내적 값

$\sum_t B_t/Q_s$: 처리 ⑤에서 용어 t에 대한 불린 값을 전체 질의어 총 수로 나눈 합

사용자가 arms라는 용어를 이용하여 검색하는 과정을 예를 들어보면 아래와 같다. 그림 2는 색인파일과 스테밍 하지 않은 파일을 보여주고 있다.

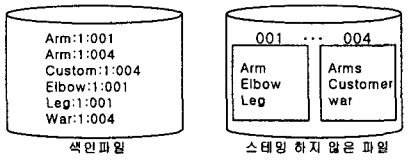

그림 2 샘플 데이터

그림 3은 일반 벡터와 본 논문에서 제안하는 시스템과의 유사도 계산 방법을 보여 주고 있다. 일반 벡터로 계산하고 문서를 순위화 하였을 경우에는 순위에 변화가 없게 되지만, 본 논문에서 제안하는 방법으로 유사도를 계산하면 문서의 순서는 바뀌게 된다.

용어 가중치 : 001, 004
TF*IDF = 1*log(4/2) = 0.3010
TF*IDF = 1*log(4/2) = 0.3010
유사도 계산

$$SIM(arm, 001) = \frac{\log(4/2) \times 1 \times \log(4/2)}{\sqrt{(\log(4/2)^2} \times \sqrt{\{1 \times (\log(4/2)\}^2}} = 1$$

$$SIM(arm, 004) = \frac{\log(4/2) \times 1 \times \log(4/2)}{\sqrt{(\log(4/2)^2} \times \sqrt{\{1 \times (\log(4/2)\}^2}} = 1$$

제안 시스템

$$SIM(arm, 001) = \frac{\log(4/2) \times 1 \times \log(4/2)}{\sqrt{(\log(4/2)^2} \times \sqrt{\{1 \times (\log(4/2)\}^2}} + 0/1 = 1$$

$$SIM(arm, 004) = \frac{\log(4/2) \times 1 \times \log(4/2)}{\sqrt{(\log(4/2)^2} \times \sqrt{\{1 \times (\log(4/2)\}^2}} + 1/1 = 2$$

그림 3 샘플 데이터에 대한 유사도 계산

4. 실험 및 평가

텍스트 문서는 태그로 된 정보로 구성이 되어 있는 데이터에서 <TEXT>와</TEXT>사이의 내용을 추출하여 문서를 생성하였다. DVL/Verity[8]의 불용어 426개를 사용하였으며 코퍼스의 크기를 줄이기 위하여 Porter 스테밍 알고리즘[9]을 사용하여 색인 파일을 구성하였다. 색인된 파일의 구성은 단어:TF:문서 순으로 되어있으며 총 702개의 문서로 된 계층 구조로 되어 있다.

TIPSTER_VOL_2의 문서 코퍼스를 실험 대상으로 사용하였고 질의어는 Tipster Topic 101~150 중에서 101, 102, 103, 104, 149, 150을 사용하였다. WSJ와 AP의 문서 코퍼스 154,431건을 대상으로 R-정확률에 대한 실험을 하였다.

4.1 평가 방법

R-정확률은 개별 적인 사용자 질의어에 대한 검색 성능을 관찰 할 수 있고 모든 질의어에 대한 R-정확률의 평균도 구할 수 있다는 장점이 있다. 따라서 본 논문에서 제안한 시스템의 성능을 평가하기 위하여 R-정확률 척도를 이용하여 시스템의 성능을 평가한다. R-정확률에서 R은 적합 문서를 의미하며, 식 (3)과 같이 상위 R개의 문서들에 포함되어 있는 적합 문서들의 비율로 정의된다.

$$R-정확률 = \frac{상위R개의문서들중에서적합문서의수}{R} \quad (3)$$

예를 들면, 표 1에서 Tipster Topic 질의어 150번에 대한 WSJ의 적합 문서의 총 수는 117개가 존재하며, 상위 100위까지 검색된 적합 문서들의 수는 21개이기 때문에, R-정확률은 21/117 = 0.179487로 계산된다. 학술대회 TREC에서는 상위 100위까지 정확률을 계산하며, 적합 문서 101부터는 검색되지 않았다고 가정하고 R-정확률을 계산한다. 따라서 나머지 문서들의 정확률은 0으로 계산된다.

표 1 질의어 150번에 대한 R-정확률

	WSJ		AP	
	일반벡터	시스템	일반벡터	시스템
TOP 5	0.008547	0.017091	0.000000	0.034188
TOP 10	0.008547	0.042735	0.017094	0.034188
TOP 15	0.008547	0.051282	0.017094	0.034188
TOP 20	0.008547	0.059829	0.025641	0.042735
TOP 30	0.025641	0.068376	0.034188	0.068376
TOP 100	0.111111	0.179487	0.085470	0.127118

WSJ와 AP 각각의 문서에 대한 질의어 150번의 적합 문서 수는 총 118개이다. 질의어 150번에 대하여 일반 벡터 공간 모델과 본 논문에서 제안하는 시스템 모두 적합 문서가 117개가 검색되었다. 표 1에서는 적합 문서가 상위에 몇 개가 위치하고 있는지에 대하여 문서 수준에 대한 적합 문서 수와 R-정확률을 나타낸다. 본 논문에서는 상위 100위까지만 적합 문서가 검색되었다고 가정하고 계산하였다. AP문서 79,911에 대한 검색에서 상위 5위에 대해서는 일반 벡터 공간 모델을 이용하여 검색하였을 경우에는 아무 문서도 검색하지 못하였지만, 용어 가중치를 재 산정한 본 논문에서 제안하는 시스템은 상위 5위안에 관련 문서가 4개가 검색이 되었다.

질의어 6개에 대한 평균 R-정확률은 표 2와 같다. 본 논문에서 제안하는 시스템이 상위 100위에 대하여 평균 R-정확률이 일반 벡터 모델보다 최고 21%의 좋은 성능을 보였다.

표 2 질의어에 대한 평균 R-정확률

	WSJ		AP	
	일반벡터	시스템	일반벡터	시스템
TOP 5	0.001425	0.015669	0.000000	0.036001
TOP 10	0.001425	0.111823	0.002849	0.051152
TOP 15	0.001425	0.121794	0.002849	0.066304
TOP 20	0.001425	0.123219	0.004273	0.067728
TOP 30	0.004273	0.124643	0.027920	0.095933
TOP 100	0.101851	0.318376	0.062729	0.150168

5. 결론 및 향후 과제

본 논문에서는 과도한 스테밍으로 인한 정확률 저하를 개선하기 위하여 용어의 가중치를 재 산정하여 일반 벡터 모델과 비교 평가하였다. 상위 100위까지에 대한 정확률이 일반 벡터 공간 모델보다 본 논문에서 제안하는 시스템이 최고 21%의 좋은 성능을 보였다.

향후 과제로는 검색 시스템의 성능 향상을 위한 온톨로지와 색인 파일 구조에 관한 연구가 있어야 할 것이다.

참고 문헌

[1] Efthimis N. Efthumiadis, "A User-Centered Evaluation of Ranking Algorithms for Interactive Query Expansion," ACM SIGIR'93, pp. 146-159, 1993.

[2] Michael Persin, "Document Filtering for Fast Ranking," SIGIR, pp.339-348, 1994.

[3] Joon Ho Lee, Yoon Joon Lee, et al., "Ranking Documents in Thesaurus- Based Boolean Retrieval System," Information Processing & Management, Vol. 30, No. 1, pp. 79-91, 1994.

[4] C. Buckley, "Implementation of the SMART Information Retrieval System," TR 85-686, Cornell Univ., Ithaca, N.Y., May 1985.

[5] DIK L. LEE, HUEI CHUANG, "Document Ranking and the Vector-Space Model," IEEE, pp. 67-75, 1997.

[6] G. Salton and M.J. McGill, "Modern Information Retrieval," McGraw-Hill, New York, 1983.

[7] G. Salton, E. A. FOX and H. WU, "Extended Boolean Information Retrieval," ACM, Vol. 26, pp.1022~1036, 1983.12.

[8] Defense Technical Information Center, DVL/Verity Stop Word List. http://dvl.dtic.mil/stop_list.html, 2002.

[9] Porter, M. Porter Stemming Algorithm. http://www.tartarus.org/~martin/PorterStemmer/.

[10] National Institue of Standards and Technology, TREC home page. http://trec.nist.gov/, 2003.