이웃한 어절간의 위치 정보를 이용하여 KRISTAL2000 DBMS 검색 성능 향상

김광영[0] 서정현 최성필
한국과학기술정보 연구원
{kykim[0], jerryseo, spchoi}@kisti.re.kr

Using the Information of Location the Improvement of KRISTAL2000 DBMS Retrieval System

Kwang-Young Kim[0] Jerry - Seo Cho Sung Pil - Choi
Group for Intelligent Information System, Korea Institute of Science and Technology Information

요 약

인터넷의 발달과 인터넷 이용자수의 급격한 증가로 정보 검색 시스템의 필요성이 커지고 있다. 또한 대용량의 문서에서 사용자가 원하는 정보를 정확하게 찾기가 점점 어려워지고 있다. 대부분의 사용자들이 입력한 질의어에 대해서 이웃한 단어를 찾아주기를 원하는 사용자가 많이 있다. 본 논문에서는 KRISTAL2000 DBMS을 이용하여 이웃하는 어절한의 위치 정보를 이용하여 다양한 가중치 방법에 대해서 실험하고 그 결과 가장 우수한 가중치 계산 방식을 적용하여 KRISTAL2000 DBMS의 성능을 향상 시키도록 하였다.

1. 서 론

인터넷의 발달과 인터넷 이용자 수의 증가로 인하여 정보 검색 시스템의 필요성이 증가하고 있다. 사용자가 원하는 정보를 정확하게 찾기가 점점 어려워지고 있다. 그리고 검색 대상 문서의 수가 급격히 증가함에 따라 검색 결과 또한 상한단 양으로 사용자가 원하는 정보인지를 쉽게 판단하고 확인하기가 어렵다. 대부분의 사용자이 입력하는 어절에 대해서 이웃하는 어절을 검색하기를 원하는 사용자도 있고 단순히 입력하는 사용자들도 있다.

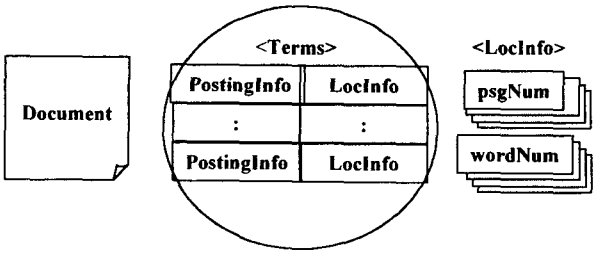
본 논문에서는 사용자가 입력하는 어절에 대해서 이웃한 어절을 검색 처리를 중점으로 두고 KRISTAL 2000을 이용하여 근접도 연산에 대해서 새로운 가중치 방법을 개선하여 서로 가까운 어절에 대해서 가중치를 추가 방법에 대해서 연구를 하였다.

KRISTAL2000 시스템에서는 한국어에 대해서는 근접도 연산자 Within을 사용하며 영어권 언어에 대해서는 Near 연산자를 사용한다.

2. 근접도 연산 처리

문서 내에서의 색인어의 위치는 색인어의 출현 빈도와 함께 검색의 정확도를 높이는 중요한 요소로 작용할 수 있다.[1] 특히, 구절 검색과 같이 정확한 어구(Phrase)를 검색하고자 할 때는 색인어의 위치 정보가 빈도보다 더 중요하다. 본 논문에서는 색인어의 문서 내 위치 정보를 포스팅 파일로 구성하여 색인어의 위치 정보를 검색에 이용하고자 한다.

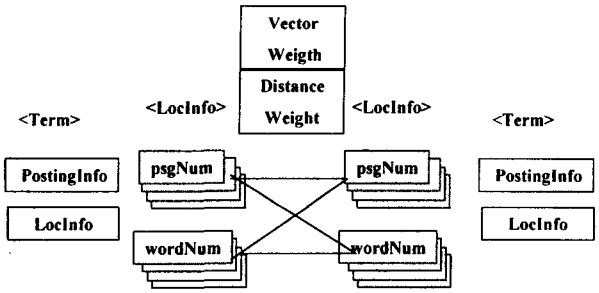
본 논문에서는 근접도 연산 처리를 위해서 각 Term마다 포스팅 정보와 위치 정보를 가지고 있다. [그림 1]에서와 같이 문서에서 색인 처리하여 Term마다 포스팅 정보와 LocInfo인 위치 정보를 가지고 있다. 이 위치 정보를 이용하여 Term과 Term사이의 위치 정보를 계산을 한다.



[그림 1] 포스팅 정보와 위치 정보

본 논문에서는 [그림 2]와 같이 포스팅의 위치 정보를 이용하여 색인어 간의 위치 정보를 계산하여 Vector모델로 계산된 가중치에 추가하는 방법을 사용하였다.

문서 내의 색인어 간 어절의 거리가 사용자 질의문의 색인어간 어절 거리와 같은 문서의 가중치를 높여주는 방식을 사용하여 가중치를 계산하였다.



[그림 2] 근접도 연산처리

3. 위치 정보를 이용한 검색

문서 내의 색인어 간 어절의 거리가 사용자 질의문의 색인어간 어절 거리와 같은 문서의 가중치를 높여 주기 위해서 아래의 [표 1]과 같은 식을 사용하였다.

[표 1] 위치 정보 계산식

식1. $Sim(Doc) = Weight_{Vector}(Doc) + \alpha$

식2. $Sim(Doc) = Weight_{Vector}(Doc) \times \frac{\alpha}{Distance}$

식3. $Sim(Doc) = Weight_{Vector}(Doc) \times (1.0 + \frac{\alpha}{log(1.0 + Distance)}$

$Distance = Term_b - Term_a$

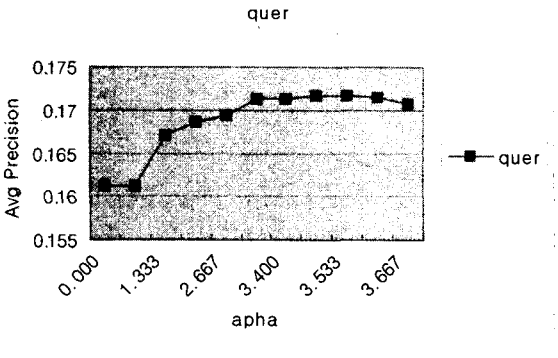
$Weight_{Vector}(Doc)$는 Term에 대한 가중치 계산인 Vector 모델로 가중치를 계산 처리한 것이다. Distance는 Terms간의 최소 거리를 선택한 것이다.

식1은 일반 Vector모델을 이용하여 가중치를 계산 한 후에 한 문장이 보통 15색인어로 구성되었다고 가정하고 일정한 상수값를 더하는 방식을 사용하였다. 식2는 일반 Vector모델을 이용하여 가중치를 계산한 후에 거리에 반비례하여 가중치를 곱하는 방식을 이용하였다. 식3은 일반 Vector모델을 이용하여 가중치를 계산 한 후에 α /log(1+Distance)의 가중치를 곱하였다. 이웃한 어절간의 근접도 가중치 처리를 본 논문에서는 위의 3가지 식을 이용하여 가장 성능이 좋은 것을 실험을 하였다.

4. 실험

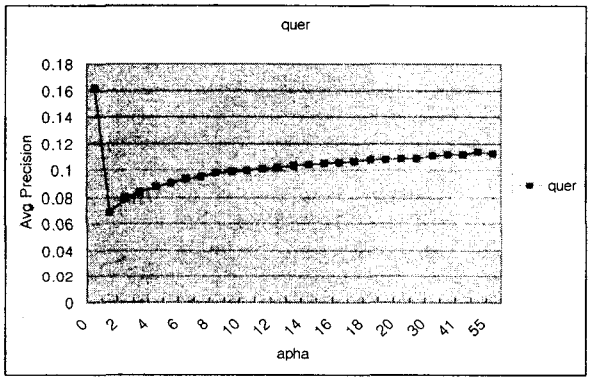
본 논문에서는 [표-1]를 이용하여 가중치를 실험을 HANTEC V2.0를 이용하여 실험을 하였다. Distance는 두 색인어간의 거리임으로 본 논문에서는 α값이 근접도 가중치에 많은 영향을 줌으로 α값을 변경하면서 실험을 하였다. 평가는 L2.rel를 이용하여 평균 Precision값을 구하였다.

식1. $Sim(Doc) = Weight_{Vector}(Doc) + \alpha$



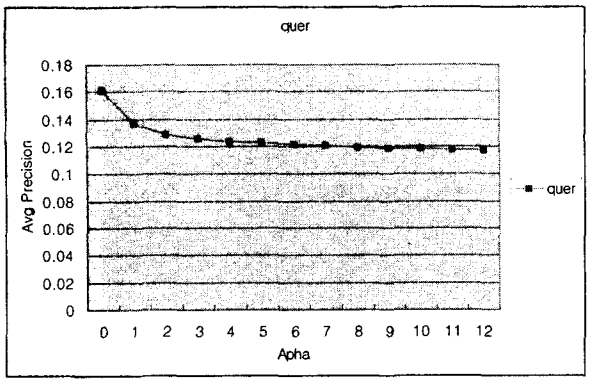
[그림 3] 식1에 대한 근접도 실험

식2. $Sim(Doc) = Weight_{Vector}(Doc) \times \frac{\alpha}{Distance}$



[그림 4] 식2에 대한 근접도 실험

식3. $Sim(Doc) = Weight_{Vector}(Doc) \times (1.0 + \frac{\alpha}{log(1.0 + Distance)}$



[그림 5] 식3에 대한 근접도 실험

식1에 대해서 실험한 결과 [그림 3]은 한 문장이 15자 정도로 구성되었다고 가정하고 가까운 거리에 상관없이 일정한 상수 값(α=3.467)을 더 했을 때 다른 식2,3보다 평균 Precision(0.1717)이 높게 나왔다.
식2에 대해서 실험한 결과 [그림 4]는 기본 벡터 가중치(0.1613)보다 가중치를 더 한 경우가 더 낮게 나타났다.
식3에 대해서 실험한 결과 [그림 5]그림은 기본 벡터 가중치(0.1613) 가중치를 더 한 경우가 더 낮게 나타났다.

5. 결론

본 논문에서는 색인어 간의 근접도 연산을 적용하여 검색 성능을 향상시키기 위해서 다양한 방식 적용하여 실험을 하였다. 실험 결과에서 볼 수 있듯이 식1에 서 가장 높은 값(0.1717)을 보였는데, 보통 한 문장의 길이를 15색인어로 구성되었다고 가정한다. 그리고 근접도 가중치 한 문장 안에서 해당될 때 일정한 값을 더 해줄 때가 가장 높은 결과를 나타내는 것을 볼 수 있다.

그러므로 본 논문에서는 일반적인 벡터 가중치와 이웃한 어절의 가중치를 추가함으로써 벡터 가중치만을 사용하는 것보다 성능이 향상됨을 알 수 있었다.

참고문헌

[1] 박대원, "용언 색인을 적용한 한국어 정보 검색시스템의 검색효율 향상" 석사학위논문 2000년 8월

[2]Park, Y. C., Choi, K. S., "*A Korean Compound Noun Retrieval Model Using Statistical NounPattern Categorization*," ICCPOL-97, pp. 361-365, 1997.

[3] 이준호, 최광남, 한현숙, "정보 검색 연구를 위한 KRIST 테스트 컬렉션의 개발" 정보관리학회지, 제2권 제2호 pp225-232, 1995

[4] 이석훈, 맹성현, 김지영, 서정현, 김현, "정보검색 평가체제 구축을 위한 HANTEC 테스트 컬렉션의 패키징" KOSTI2000 pp31-48