효율적인 질의응답시스템 개발을 위한 BM25기반의 단락 검색 시스템

임희석, 이영신, 임해창

요 약

본 논문은 문서 단위 보다 작은 단락 단위의 검색 시스템을 사용하는 효율적인 질의 응답 시스템 개발을 위하여 문서 검색에서 성능이 검증된 Okapi 시스템의 BM25 알고리즘을 응용한 단락 검색 시스템을 제안하고, 단락 검색 시스템의 성능을 분석하고자 한다. 100만 건의 문서로 구성된 TREC Q&A track 테스트 컬렉션을 색인에 사용하고 TREC Q&A track 질의 집합 중 1 ~ 100번까지의 질의를 사용하여 실험한 결과 재현율이 100%가 되기 위해서 문서 검색은 약 12만 문장을 검색해야 하는 반면, 단락 검색에서는 문서 검색의 약 1/70인 1700문장만으로도 100%의 재현율을 얻을 수 있음을 확인하였다.

A BM25 based Passage Retrieval System for Developing an Efficient Question and Answering System

Heui Seok Lim, Yong Shin Lee, Hae Chang Rim

ABSTRACT

This paper proposes a passage retrieval system based on Okapi's BM25 for developing an efficient QA system and evaluates performances of the passage retrieval system. The test collection of TREC Q&A track which is composed of about one million documents was indexed and a hundred queries of TREC Q&A track are used as testing queries. The experimental results shows that the proposed passage retrieval system can reach to 100% recall rate by searching in only 1700 sentences while the conventional document retrieval system have to search about 120 thousands sentences which are about 70 times more than the proposed passage retrieval system.

keywords : question and answering system, information retrieval, TREC

1. 서 론

질의 응답 시스템은 사용자의 질의와 관련된 문서를 검색하는 정보 검색 시스템과는 달리 사용자의 질의에 대한 답변이 될 수 있는 정답을 문서 집합 내에서 탐색하여 사용자에게 제시해 주는 시스템이다[1]. 일반적으로 질의 응답 시스템은 사용자의 질의에 관련된 문서를 검색하는 후보 검색 단계(candidates retrieval phase)와 검색된 문서 내에서 정답을 생성하는 정답 추출 단계(answer extraction phase)로 구성된다.

정답 추출 단계는 구문 분석 또는 의미 분석 등과 같은 고급의 언어 처리 기술을 사용하여 사용자의 질의에 적합한 정답을 추출하는 단계이다. 따라서 정보 검색 시스템에서 사용되는 색인 가능한 기본적인 정보 이외에도 색인 할 수 없는

논문접수:2003년 9월 15일, 심사완료:2003년 10월 16일
본 연구는 학술진흥재단의 신진교수연구과제(2001-003-E00271)의 지원에 의한 것임.

다양한 구문 정보 혹은 의미 정보들을 사용하여 정답임을 판별해 내는 분석 작업이 수행되어야 한다. 색인 되어있지 않은 정보들을 이용하는 이러한 특성으로 인해 정답 추출 과정이 모든 문서에 대해 일괄적인 적용되는 방법은 사용되기 어렵다. 이로 인하여 질의 응답 시스템에서는 본격적인 정답 추출 작업을 수행하는 전 단계로 검색 시스템을 사용해서 정답을 포함하고 있을 가능성이 있는 문서들을 선별해내는 후보 검색 단계를 수행한다. 즉 후보 검색 단계는 정답 추출 단계가 적용될 후보를 찾아내는 전처리 단계이다. 따라서 후보 검색 시스템의 재현율, 정확도 그리고 검색 결과량은 질의 응답 시스템 전체의 재현율, 정확도 그리고 성능에 결정적인 영향을 미친다.

2. 관련 연구

일반적으로 질의 응답 시스템에서 사용되는 후보 검색 방법은 문서 전체를 검색 대상으로 하는 문서 검색(document retrieval) 방법과 문서 내에 존재하는 문단이나 문장 등과 같은 텍스트 일부분을 검색 대상으로 하는 단락 검색(passage retrieval) 방법으로 구분할 수 있다[2, 3, 4, 5, 6].

문서 검색을 수행하는 대표적인 시스템들로는 [2], [3], [4] 등이 있다. [2]의 경우엔 불린 모델의 PRISE 검색 엔진을 이용하여 문서 검색을 수행한 후 검색된 문서들 중에서 제한된 window size 이내에 해당 불린 질의를 만족하는 단락들이 있는지 검사를 수행하여 단락을 filtering해 낸다. 이렇게 추출된 단락들의 수가 너무 많거나 혹은 너무 적으면 해당 불린 질의를 좀더 강화시키거나 완화시키며 문서 검색을 다시 수행하여 적절한 수의 단락들이 추출될 때까지 문서 검색과 filtering 과정을 반복 수행한다.

[3]의 경우엔 2차에 걸친 문서 검색을 수행하는데, 첫 번째 검색 단계에서는 25단어 이내에 모든 키워드를 포함하고 있는 문서를 검색한다. 이렇게 까다로운 조건을 만족하는 문서는 일반적으로 그 양이 아주 작으므로 이를 보완하기 위하여 두 번째 검색 단계에서는 키워드 빈도와 가중치를 사용하는 일반적인 문서 검색을 수행하여 1, 2 단계 도합 60개의 문서를 검색해 낸다. 이렇게 검색된 문서들을 문장 단위로 분리를 한 후 각각의 문장들을 대상으로 개체명 인식 등을 통하여 정답 후보를 포함하고 있는 문장들을 filtering한다.

[4]의 경우엔 불린 모델의 검색 엔진을 통하여 문서 검색을 수행한 후 검색된 문서들을 문장 단위로 분리를 한 뒤 개체명 인식 등을 통하여 정답 후보를 포함하고 있는 문장들을 filtering하고, 추출된 각 문장들을 여러 가지 휴리스틱을 사용하여 순위화 한 뒤 최종적으로 상위 300개의 문장들을 추출해낸다.

문서 검색을 수행하는 시스템들의 공통적인 특징은 검색된 문서들에 대해서 자신들만의 독특한 filtering 기법을 사용한다는 것이다. 즉, 문서 내에서 정답과 관련성이 있는 부분은 문서 전체가 아니라 문서의 일부분이기 때문에 문서에서 필요한 부분들만을 뽑아내어 사용한다. 이는 문서 검색 방법이 정답을 찾기 위해서 검색하는 양이 불필요하게 너무 많기 때문에 질의 응답 시스템에서 문서 검색 방법을 사용하려면 이를 줄여주는 부가적인 방법이 반드시 필요함을 의미한다.

단락 검색 방법을 사용하는 대표적인 시스템들로는 [5, 6]이 있다. [5]에서는 본 검색을 수행하기 이전에 웹 백과사전을 대상으로 검색을 수행하여 질의 확장을 한 후 확장된 질의를 이용하여 본 검색을 수행한다. 본 검색 단계에서는 문서 집합 내의 모든 문서들을 대략 200개의 단어로 이루어진 단락들로 분할하는 전처리 과정을 거쳐 문서 단위가 아닌 단락 단위로 색인이 된 정보를 사용하여 최종적으로 70개의 단락을 검색한다.

[6]에서는 문서의 임의의 범위를 대상으로 해당 범위의 길이, 포함하고 있는 키워드의 수 등을 이용하여 스코어를 계산하여 검색한 후 상위 10개에 대해 각각 그 중앙을 기준으로 200 단어 크기의 단락을 추출한다.

단락 검색 방식들은 문서 검색 방법에 비해 filtering 등과 같은 추가적인 후처리 과정 없이 질의 응답 시스템에서 사용할 수 있는 적당한 양의 단락들을 직접 추출해 낼 수 있다는 장점이 있다.

이와 같이 질의 응답 시스템에서 사용되는 검색 시스템은 정답 추출 단계의 정답 탐색 범위를 줄여주는 중요한 역할을 수행한다. 뿐만 아니라

질의 응답 시스템은 검색 시스템의 결과로부터 정답을 추출하기 때문에 질의 응답 시스템의 성능은 검색 시스템의 성능에 의존적이다. 즉, 검색 시스템에서 정답을 찾지 못한다면, 아무리 성능이 우수한 질의 응답 시스템이라 할지라도 정답을 찾을 수 없게 된다. 따라서 효율적인 질의 응답 시스템 개발을 위해서는 후보 검색 단위를 단락으로 사용하여 정답 추출 단계에서의 부하를 최소화시키면서도 정답이 포함된 부분을 모두 정확히 후보로 생성할 수 있는 높은 재현율과 정확률을 갖는 후보 검색 방법을 요구한다.

전통적으로 문서를 검색 단위로 사용하는 정보 검색에 관한 연구는 상당히 진행된 반면에 효과적인 단락 검색 방법에 대한 연구는 매우 미흡한 실정이다. 이에 본 논문은 효율적인 한국어 질의 응답 시스템 개발을 위하여 문서 검색 방법에서 이미 성능이 검증된 Okapi 시스템의 BM25 알고리즘[8]을 단락 검색에 적용할 수 있는 방법을 제안하고, 이를 이용한 질의 응답 시스템의 효율 향상을 실험을 통하여 보이고자 한다.

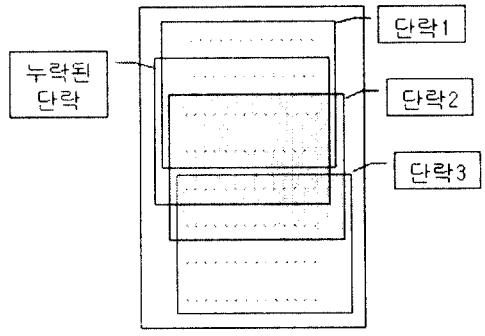
3. 단락검색시스템

3.1. 단락의 구성

단락 검색에서 단락은 n개의 문장으로 그 크기가 정의한다. 즉, 고정 길이 단락 검색은 문서 내에 존재하는 연속된 n개의 문장에 대해 스코어를 계산하고 해당 부분만을 문서에서 추출한다.

단락의 크기에 대한 정의 이외에도 단락 검색을 수행함에 있어 고려되어야 할 또 다른 사항이 있는데, 바로 단락이 결정되는 시점이다. 단락을 결정하는 시점은 색인을 하는 시점과, 검색을 하는 시점 두 가지 경우가 있다.

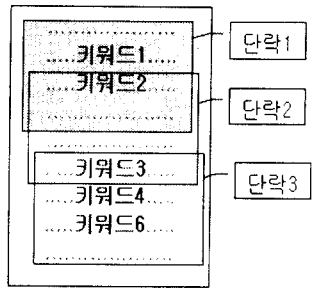
색인시 단락을 결정하는 경우엔 일반적으로 단락의 반이 다른 단락과 중첩되도록 문서에서 단락들을 추출하여 색인한 뒤, 이러한 단락들을



[그림 1] 색인된 단락 및 색인에서 누락된 단락들

독립된 작은 문서로 간주하여 검색을 진행한다[7]. 이 방식은 기존의 문서 검색 방법을 그대로 단락 검색에 적용할 수 있다는 편의성이 있기 때문에 일반적으로 많이 사용되는 방법이지만, 중첩된 단락을 사용했음에도 불구하고 고려하지 못하는 많은 단락들이 존재하는 단점이 있다. [그림 1]은 이와 같이 색인 시 단락을 결정할 경우 색인에서 누락되는 단락이 발생하는 모습을 보여준다.

검색 시 단락을 결정하는 경우엔 키워드들의 출현 위치 정보를 색인시점에 미리 저장한 뒤 검색 시 문서 내에서 키워드가 출현하는 위치를 기점으로 단락을 추출한다. 즉, 키워드가 출현한 문장을 포함하고 있는 연속된 n개의 문장을 단락으로 추출한다. 이 방식은 색인 시 단락을 결정하는 방법에 비해 계산량이 늘어나는 단점이 있지만, 존재할 수 있는 모든 단락들을 고려할 수 있다는 장점이 있다. [그림 2]는 검색 시 단락을 결정하는 모습을 보여준다. 본 논문은 단락을 결정하는 시점으로 검색 시점을 선택하여 비록 계산량이 늘어나지만 단락이 생성될 수 있는 가능한 모든 경우를 고려할 수 있도록 하였다.



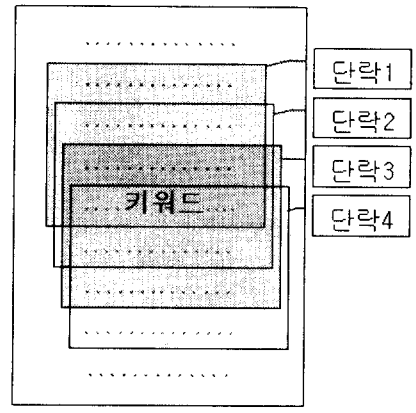
[그림 2] 중첩된 단락을 처리한 후의 모습

3.2. 중첩된 단락처리

검색 시점에 단락을 결정하는 경우에는 비슷한 내용을 포함하고 있는 중첩된 단락들이 다수 개가 검색되는 현상이 발생하게 된다. 특히 특정 문장 내에 다수의 키워드들이 밀집해 있는 경우 그 문장을 포함하고 있는 모든 중첩된 단락들은 모두 높은 순위로 검색이 된다. 비록 이들이 별개의 단락이긴 하지만, 단락 안에 포함된 정보는 거의 동일한데, 이들이 중복되어 높은 순위로 검색이 되는 것은 동일 문서 혹은 타 문서의 다른 부분에서 추출되는 유용한 단락들이 선택될 기회를 박탈하는 부정적인 작용을 하게 된다. 본 논문은 이러한 단점을 보완하기 위해, 검색 시 키워드가 출현한 각각의 문장에 대해서 해당 문장을 기준으로 추출 가능한 단락들 중에 스코어가 가장 높은 단락 하나만을 추출하도록 제한을 가하도록 하였다. [그림 3]은 중첩된 단락을 처리하고 난 후 최종적으로 추출되는 단락들의 모습을 일례로 보여주고 있다.

3.3. BM25기반의 단락가중치계산

단락과 사용자 질의와의 유사도 계산은 TREC[1]에서 이미 그 성능이 검증된 Okapi 시스템의 BM25를 단락 검색에 적합하도록 수정하여 사용한다. 기본적으로 BM25 알고리즘은 문서 단위의 알고리즘이므로 단락 검색에 적합하도록 파라미터들의 의미를 수정한 식(1)을 사용한다[8]. 식(1)과 BM25의 파라미터들 간의 차이는 [표 1]과 같다.



[그림 3] 검색시점에 추출되는 단락들

$$score = \sum_{t \in Q} \frac{(k_1+1)tf}{K+tf} \cdot \frac{(k_3+1)qtf}{k_3+qtf} \log \frac{N'-n'+0.5}{n'+0.5} \quad (1)$$

$$where,\quad K = k_1((1-b)+b\frac{pl}{avpl}),$$
$$k_1 = 1.2,\quad b = 0.75,\quad k_3 = 1000$$

제안한 시스템은 색인 시점에 모든 단락들을 구성하여 색인하는 것이 아니라 검색 시에 동적으로 단락을 구성하는 방식을 사용하기 때문에 식(1)에서 사용되는 파라미터들 중 전체 단락들의 수 N′ 및 해당 키워드를 포함하고 있는 단락들의 수 n′에 대해서는 그 정확한 값을 알아낼 수 없다. 따라서 N′ 및 n′에 대해서는 근사값을 사용하는데, 근사값을 구하는 방법은 다음과 같다.

전체 문서 집합에서 문장의 총 수를 S, 해당 키워드를 포함하고 있는 문장의 수를 s라 하자. 단락의 길이가 m 문장으로 고정되어져 있다고 하면, 한 문서에서 생성 가능한 모든 단락의 수는 (문서의 문장 개수 + 1 - m) 개다. 따라서, 전체 문서 집합에서 생성 가능한 단락의 총 수 N′은 식(2)와 같이 근사될 수 있다.

[1] NIST와 DARPA의 지원을 받는 세계적인 정보검색 평가대회이며 1992년 TIPSTER의 text program의 한 부분으로 시작되었다.

[표 1] BM25에서 변형된 파라미터들

BM25		식(1)	
N	전체 문서의 수	N'	전체 단락의 수
n	키워드를 포함하고 있는 문서의 수	n'	키워드를 포함하고 있는 단락의 수
dl	문서의 길이	pl	단락의 길이
$avdl$	문서의 평균 길이	$avpl$	단락의 평균 길이

$$\begin{aligned} N' &\approx (\text{문서당 평균 문장 개수} + 1 - m) \\ &\quad \times \text{전체 문서 개수} \\ &= (\frac{S}{N} + 1 - m) \times N \\ &= S + N - m \cdot N \end{aligned} \tag{2}$$

식(2)에서 알 수 있듯이, N'은 m이 커질수록 N에 가깝게, m이 작아질수록 S에 가깝게 근사된다. 키워드를 포함하고 있는 단락의 수 n'은 실제로 문서 내에서 키워드가 출현하는 위치에 따라 그리고 키워드들 간의 간격에 따라 유동적인 값을 갖게 되므로 그 값을 직접 근사하기는 어렵다. 하지만, n' 역시 m이 커질수록 n에 가깝게, m이 작아질수록 s에 가깝게 근사되는 특성이 있으므로 동일한 특성을 지니는 N'을 이용하여 다음과 같이 간접적으로 그 값을 근사할 수 있다.

전체 문서 집합을 하나의 덩어리라고 가정하자. 그 덩어리를 문서 단위로 분할했을 때, 분할된 조각들의 개수는 N이고 그 조각들 중 키워드를 포함하고 있는 조각은 n개이다. 전체 문서 집합을 문장 단위로 분할하면, 분할된 조각들의 개수는 S이고 그 조각들 중 키워드를 포함하고 있는 조각은 s개이다. 그렇다면 전체 문서 집합을 고정된 길이의 단락 단위로 분할했을 경우, 분할된 조각들의 개수가 N'일 때 키워드를 포함하고 있는 조각은 몇 개인지가 우리가 구하고자 하는 값이다. 이를 좌표로 바꾸어 표현해보면, 분할된 조각들의 개수를 x, 키워드를 포함하고 있는 조각들의 개수를 y라 하면, (N, n), (S, s) 두 점이 미리 주어지고 x값이 N'일 경우 y값을 구해내는 문제로 볼 수 있다. 이 y값을 구하는 가장 쉬운 방법은 보간법(interpolation)을 사용하는 것이다. 이와 같이 두 점이 주어진 경우엔 단순한 선형 보간법(linear interpolation)을 사용하여 n'을 근사할 수 있다. 이 때, n'과 근사값 간의 오차를 줄이기 위해 사용할 수 있는 정보가 하나 더 있는데, 바로 전체 단어의 수 T와 키워드의 출현수 t이다. 전체 문서 집합을 단어 단위로 분할할 경우, 분할된 조각들의 개수는 T이고 키워드를 포함하고 있는 조각은 t개이다. 따라서, (N, n), (S, s), (T, t) 세 점과 x값 N'이 주어진 경우에 대해 보간법을 사용하여 n'을 구하면 된다. 본 논문에서는 식(3)과 같이 Lagrange 보간 다항식을 사용하여 n'을 근사한다.

$$\begin{aligned} n' &= \frac{(N'-S)(N'-T)}{(N-S)(N-T)} n \\ &+ \frac{(N'-N)(N'-T)}{(S-N)(S-T)} s \\ &+ \frac{(N'-N)(N'-S)}{(T-N)(T-S)} t \end{aligned} \tag{3}$$

최적의 검색 성능을 얻기 위해서는 여러 실험을 통해 k1, b, k3 파라미터의 값을 설정해야 하겠지만, 본 논문은 단락의 특성에 따른 검색 성능의 추이를 살피기 위하여 식 (1)에 제시된 값을 그대로 사용한다.

4. 실험 및 평가

본 논문에서 제안한 시스템은 TREC Q&A track 테스트 컬렉션[2]에 적용하여 평가하였다. Q&A track 컬렉션은 대략 100만 건의 문서로 이

[2] TREC QA 테스트 컬렉션은 TIPSTER 및 TREC 컬렉션 중 다음과 같은 신문 및 뉴스 기사들로 구성되어 있다.
AP newswire (disks 1-3)
Wall Street Journal (Disks 1-2)
San Jose Mercury News (Disk 3)
Financial Times (Disk 4)
Los Angeles Times (Disk 5)
Foreign Broadcast Information Service (Disk5)

[표 2] 재현율 100%일 때의 문서 검색 및 단락 검색의 검색량

	doc	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p12	p15	p17	p22
rank	119142	12010	1719	3299	4356	8925	14732	20566	35180	47079	90785	27891	50644	40711	32716

루어져 있는데. 문장 단위의 색인을 위하여 모든 문서에 대해 문장을 구분하는 전처리 작업을 수행하였고, 불용어를 제외한 모든 단어들에 대해 스테밍(stemming) 처리를 한 뒤 색인하였다. 실험에 사용된 질의는 TREC Q&A track 질의 집합[3] 중 1 ~ 100번 까지 총 100개의 질의들을 사용하였다.

실험 결과의 평가 방법은 전체 질의 중 제한된 검색량 내에서 정답을 찾은 질의의 개수, 즉, 식(5)와 같은 재현율[4]을 평가 척도로 사용하였다. 그리고 크기가 서로 다른 단락들에 대한 비교 평가를 위해서 검색 결과의 제한량은 단락 단위가 아닌 문장 단위로 순위화 하여 평가하였다. 즉, 검색된 단락들을(또는 문서들을) 순위화하여 배열한 m개의 단락(또는 문서)에서 n개의 문장을 검색 결과로 추출하여 그 중에 정답이 포함되어 있는지를 평가하였다.

$$\text{재현율} = \frac{\text{정답이 찾아진 질의의 수}}{\text{전체 질의의 수}} \times 100 \quad (4)$$

실험은 문서 검색과 단락 검색에 대하여 각각 두 가지의 평가를 수행하였다. 첫째, 제한된 검색량 내에서의 재현율이다. 즉, 상위 n개의 문장을 검색하였을 때 정답을 찾은 질의의 비율에 대한 평가이다. 기존의 단락 검색 시스템에서 사용한 검색량이 대략 1000문장 내외임을 감안하여 본 논문에서는 100문장에서 1000문장까지 단계별로 재현율을 분석하였다. 둘째는, 재현율이 100%일 때의 검색량이다. 즉, 모든 질의에 대해 정답을 찾기 위해서 검색해야하는 최소 검색량에 대한 평가이다. 단락 검색에 대해서는 테스트 컬렉션의 문서당 평균 문장수가 대략 22문장임을 고려할 때, 단락의 크기를 1문장에서부터 22문장까지 변화시키면서 총 22가지의 크기의 단락에 대하여 각각 실험하였고, 최고 1000 문장까지의 검색량에 대해 실험을 하였다.

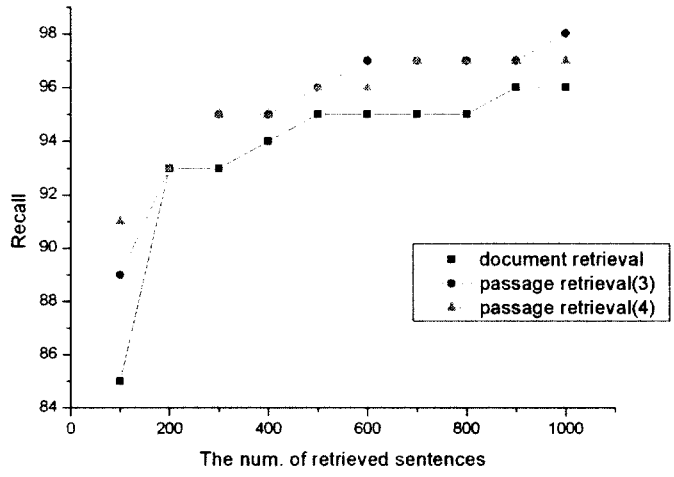
실험 결과 전체적으로 단락의 크기가 큰 경우보다 작은 경우에 비교적 성능이 우수하게 나타났는데, 그 이유는, 단락의 크기가 커질수록 검색된 각 단락 내에 불필요한 부분들이 많이 포함되어 결과적으로 제한된 검색량 내에 불필요한 부분이 차지하는 비중이 점점 더 커지기 때문이다. 또한, 검색량이 클 경우엔 단락의 크기에 따른 성능의 차이가 그리 많지 않은 반면, 검색량이 작은 경우엔 단락의 크기에 따른 성능 차이가 커지는 경향이 있다. 즉, 검색량에 제한을 많이 받을수록 단락의 크기를 작게 하여 검색을 하는 것이 정답을 찾는데 유리함을 알 수 있다.

[그림 4]는 실험 결과 성능이 가장 우수하게 나타난 제안한 단락 검색 방법(passage retrieval 3, passage retrieval 4)과 문서 검색 방법(document retrieval)의 재현율을 나타낸 그래프로 나타낸 것이며 x축은 검색된 문장의 수를 나타내고, y축은 재현율을 나타낸다. [그림 4]의 결과를 살펴보면 문서 검색 보다는 본 논문이 제안한 단락 검색이 보다 더 높은 재현율을 보이고 있다. 이는 문서의 특정 부분에 정답이 있는지를 확인하기 위해서는 그 주변의 3, 4 문장을 살펴보는 것만으로도 정답의 유무를 대체로 확인해 낼 수 있으며 단락 검색을 이용하면 동일한 처리량으로도 높은 재현율을 얻을 수 있음을 의미하는 것이다.

[표 2]는 재현율이 100%가 될 때까지 검색을 수행했을 경우 얻어지는 검색 결과 량을 보이고 있다. 모든 정답을 다 찾기 위해서는 문서 검색의 경우엔 약 12만개의 문장을 검색해야 하는 반면

3) http://trec.nist.gov/data/qa/T8_QAdata/topics.qa_questions.txt

4) 본 논문에서 사용되는 재현율은 질의 응답 시스템에 적합한 평가방법으로서, 일반적인 정보 검색 시스템에서의 재현율(전체 정답 중 검색된 정답의 비율)과는 그 의미하는 바가 상이하다.



[그림 4] 문서 검색과 제안한 단락 검색과의 성능 비교

단락의 크기가 2인 고정 길이 단락 검색의 경우엔 약 1700여개의 문장을 검색하면 된다. 즉, 문서 검색에 비해 고정 길이 단락 검색의 결과량은 약 1/70으로 줄어듬을 알 수 있다. 이는 단락 검색을 사용하는 질의 응답 시스템은 문서 검색을 사용하는 경우보다 단지 1/70의 계산량으로 동일한 정확률을 갖는 정답을 추출할 수 있음을 나타내는 것이다.

5. 결론

본 논문은 문서 단위의 검색에서 이미 성능이 검증된 Okapi 시스템의 BM25 알고리즘을 단락 검색에 적용할 수 있는 방법을 제안하였고, 이를 이용한 질의 응답 시스템의 효율 향상 여부를 실험으로 밝혔다.

제안한 단락 검색의 경우 일반적으로 단락의 크기가 2 ~ 4문장일 경우 최적의 성능을 나타내었고, 일반적으로 단락의 크기가 큰 경우보다는 작은 경우에 검색의 성능이 좋으며, 특히 검색량에 대한 제한이 많이 가해질수록 작은 크기의 단락을 사용하는 것이 더욱 더 유리함을 보였다. 재현율이 100%가 되기 위해서 문서 검색은 약 12만 문장을 검색해야 하는 반면, 단락 검색에서는 그것의 1/70인 1700문장만으로도 100%의 재현율을 얻을 수 있었다.

문서 내에서 정답과 관련성이 있는 단락의 크기는 문서마다 그 크기가 서로 상이할 수 있다. 어떠한 문서에서는 1문장 안에 정답과 관련된 키워드들이 모두 포함되어 있을 수도 있고, 다른 문서에서는 10문장에 걸쳐 키워드들이 분산되어 있을 수 있다. 하지만 본 논문에서 제안한 단락 검색의 경우엔 단락의 길이가 고정되어 있기 때문에 이렇게 각 문서마다 관련된 부분의 크기가 서로 상이한 특성을 고려할 수 없는 단점이 있다. 즉, 단락의 크기가 큰 경우엔 불필요한 내용이 많이 포함되어지는 문서들이 존재하게 되고, 단락의 크기가 작은 경우엔 필요한 내용을 모두 포괄할 수 없는 문서들이 존재하게 된다. 이러한 문제를 극복하기 위해서는 각 문서마다 그에 적합한 크기의 단락을 동적으로 결정해 줄 수 있는 방법이 필요로 하는데, 이를 위해 가변 길이 단락 검색에 대한 추가적인 연구가 필요할 것으로 생각된다.

참고문헌

[1] Ellen M. Voorhees, Dawn Tice, "The TREC-8 Question Answering Track Evaluation", Proceedings of the 8th Text REtrieval Conference(TREC-8), 1999.

[2] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu. "FALCON: Boosting Knowledge for Answer Engines", In the Proceedings of Text REtrieval Conference (TREC-9), 2000.

[3] S. Alpha, P. Dixon, C. Liao, "Oracle at TREC 10", In the Proceedings of Text REtrieval Conference (TREC 2001), 2001.

[4] E. Hovy, U. Hermjakob, C-Y Lin, "The Use of External Knowledge in Factoid QA", In the Proceedings of Text REtrieval Conference (TREC 2001), 2001.

[5] J. Prager, J. Chu-Carroll, "Use of WordNet Hypernyms for Answering What-Is Question", In the Proceedings of Text REtrieval Conference (TREC 2001), 2001.

[6] C. L. A. Clarke, G. V. Cormack, D. I. E. Kisman, T. R. Lynam, "Question Answering by Passage Selection (MultiText Experiments for TREC-9)", In the Proceedings of Text REtrieval Conference (TREC-9), 2000.

[7] James P. Callan, "Passage-Level Evidence in Document Retrieval", In the proceedings of the 17th ACM SIGIR conference on research and development in information retrieval, 1994.

[8] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3", in the Proceedings of Text REtrieval Conference (TREC-3), 1995.

임 희 석

1992. 2 : 고려대학교 컴퓨터학과(학사)
1994. 2 : 고려대학교 컴퓨터학과(석사)
1997. 9 : 고려대학교 컴퓨터학과(박사)
1997. 9 ~ 1999. 2 : 삼성종합기술원 전문연구원
1999. 3 ~ 현재 : 천안대학교 정보통신학부 교수
관심분야 : 자연어처리, 인공지능, 인지신경과학
E-Mail : limhs@infocom.chonan.ac.kr

이 영 신

2000. 2 : 고려대학교 컴퓨터학과(학사)
2003. 2 : 고려대학교 컴퓨터학과(석사)
2003. 3 ~ 현재 : (주)엔엘피 솔루션 선임연구원
관심분야 : 자연어처리, 컴퓨터 게임
E-Mail : notenoughenergy@hotmail.com

임 해 창

1979. 2 : 고려대학교 독어독문학(학사)
1983. 2 : Missori 주립대학 전산학(석사)
1990. 2 : Texas 주립대학 전산학(박사)
1991. 3 ~ 현재 : 고려대학교 컴퓨터학과 교수
관심분야 : 자연어처리, 인공지능, 정보검색
E-Mail : rim@nlp.korea.ac.kr