질의 응답 시스템을 위한 가변 길이 단락 검색

이영신[0] 황영숙 임해창
고 려 대 학 교 컴 퓨 터 학 과
{yslee, yshwang, rim}@nlp.korea.ac.kr

Variable Length Passage Retrieval for Q&A System

Young-Shin Lee[0] Young-Sook Hwang Hae-Chang Rim
Dept. of Computer Science, Korea University

요 약

질의 응답 시스템에서 보다 정확하게 정답을 판별하기 위해서는 구문분석 혹은 의미분석 등과 같은 복잡도가 높은 분석작업이 요구되며, 이러한 질의 응답 시스템 성능의 상한을 결정하는 검색 시스템은 가급적 적은 양의 검색 결과를 내주어서 질의 응답 시스템이 처리해야 할 작업량에 대한 부담을 덜어주어야 한다. 본 논문에서는 이러한 요구를 만족시키는 검색 시스템으로 가변 길이 단락 검색 시스템(variable length passage retrieval system)을 제안한다. 제안하는 검색 시스템은 질의에 대한 정답을 포함하고 있을 가능성이 있는 텍스트 영역은 질의에 따라 그 크기가 다를 것이라는 가정으로부터 출발한다. 그러므로 문서 전체를 검색하거나 고정 길이 단락으로 나누어져 색인되어 있는 부분 문서들을 검색하는 기존의 검색 방법과 달리, 제안된 시스템은 문서에서 임의의 길이로 이루어진 단락을 대상으로 동적인 단락 검색을 수행한다. TREC QA track의 질의집합 중 1번부터 100번까지의 질의에 대해 실험을 수행한 결과, 문서 검색 시스템이나 고정 길이 단락 검색 시스템은 상위 1000개의 문장까지 검색을 하였을 때 각각 96%, 98%의 재현율을 보인 반면, 가변 길이 단락 검색 시스템은 800개의 문장 만으로도 98%의 재현율을 보이고, 900개의 문장을 검색하였을 경우 100%의 재현율을 보였다.

1. 서론

질의 응답 시스템은 사용자의 질의에 대한 답변이 될 수 있는 정답을 문서 집합 내에서 검색하여 사용자에게 제시해 주는 시스템이다. 광범위한 관점에서의 질의 응답 시스템은 옳고 그름의 여부에 따라 '예/아니오'를 답변하는 것에서부터 다양한 문서 집합 내에 존재하는 정보들을 분석하고 종합하여 복잡한 결과를 답변하는 것에 이르기 까지를 모두 포괄할 수 있지만, 본 논문에서는 TREC에서 제안하는 것과 같이 사실에 기반한 짧은 길이의 정답을 갖는 질의를 대상으로 하는 질의 응답 시스템만을 고려한다[1].

대량의 문서집합 속에서 사용자가 원하는 정보를 검색한다는 점에서 질의 응답 시스템은 정보 검색 시스템과 유사한 특징이 있다. 그러나 일반적으로 정보 검색 시스템이 사용자의 질의와 관련된 문서들을 찾는 데 반해 질의 응답 시스템은 질의에 대한 정확한 답을 찾아야 한다는 점에서 일반적인 정보 검색 시스템보다 더욱 정밀한 검색 작업이 요구된다. 즉, 정보 검색 시스템에서 사용되는 색인 가능한 기본적인 정보 이외에도 색인 할 수 없는 다양한 구문 정보 혹은 의미 정보들을 사용하여 정답임을 판별해 내는 분석작업을 수행한다. 색인 되어 있지 않은 정보들을 이용하는 이러한 특성으로 인해 모든 문서들에 대해 일괄적인 검색 작업을 수행하는 방법은 사용될 수 없고, 결국 각각의 문서에 대해 개별적으로 분석작업을 수행하여야 하는데 이 역시 현실적으로 가능하지 않다. 그래서 질의 응답 시스템에서는 본격적인 분석 작업을 수행하는 전 단계로 검색 시스템을 사용해서 정답이 있을법한 문서들을 선별해내는 검색 작업을 수행한다.

질의 응답 시스템에 사용되는 검색 시스템에서는 검색된 문서들 내에 질의와 관련된 문서가 얼마나

많이 분포하고 있는가 정확도는 얼마나 높은가 하는 것은 큰 의미가 없다. 중요한 점은 검색된 결과 내에 정답이 포함되어 있는가 하는 것이다. 이러한 관점에서 보면 가급적 많은 양을 검색해 내서 정답이 그 안에 들어있을 확률을 높이는 것이 바른 접근 방법이라 볼 수 있다. 그러나 현실적으로 주어진 환경에서 제한된 시간 내에 질의 응답 시스템이 분석해 낼 수 있는 처리량은 제한되어 있고, 분석 작업을 좀 더 깊고 다양하게 하여 정답을 보다 잘 찾고자 할수록 처리량은 점점 더 줄어들게 된다. 따라서 보다 깊고 정확하게 분석작업을 수행하는 질의 응답 시스템일수록 보다 적게 텍스트를 뽑아주는 검색 시스템이 필요하게 된다. 이는 질의 응답 시스템을 위한 검색 시스템이 가급적 적은 양의 검색 결과 내에서 정답을 찾아내야 하는 요건을 갖추어야 함을 의미한다.

보편적으로 검색 결과의 양은 검색 순위를 어떻게 제한하는지에 의해 결정된다. 많은 양의 검색 결과를 사용하고 싶으면 낮은 순위까지, 적은 양을 사용하고 싶으면 높은 순위까지의 검색 결과를 사용하면 된다. 하지만 검색 결과의 양에 영향을 미치는 다른 요인으로서 검색 대상의 크기를 고려해 볼 수 있다. 일반적으로는 문서 전체가 검색의 대상으로서 사용되지만, 문서 내에 존재하는 각각의 문단이나 몇 개의 문장 등과 같은 일부의 텍스트 영역[1](단락) 역시 검색의 대상이 될 수 있다. 따라서 동일한 순위까지를 검색 하더라도 검색한 대상이 문서인지, 문단인지, 혹은 문장인지에 따라 검색된 양은 크거나 작게 된다. 따라서, 정답을 포함하면서도 가급적 적은 양의 검색 결과를 사용하는 것은 검색 대상의 크기와 제한된 검색 순위간의 효율적인 조합 문제로 접근해 볼 수 있다. 가령, 문서를 검색 대상으로 상위 10위 까지를 검색한다든지, 문단을 검색 대상으로 상위 100위 까지를 검색한다든지 하는 조합들이 있을 수 있겠다.

이에 본 논문에서는 단락의 특성에 따른 각각의 단락 검색 시스템들을 구현하고, 실험을 통해 각 단락 검색 시스템의 검색 성능을 분석한다. 그리고 실험 결과로 가변 길이 단락 검색(variable length passage retrieval)이 앞에서 언급한 질의 응답 시스템을 위한 검색 성능 요건을 충족시키는 적합한 방법임을 제시한다.

2. 관련 연구

질의 응답 시스템에 사용된 기존의 검색 시스템들을 살펴보면 질의의 구성 방식이나 스코어 계산 방식, 적합성 피드백 방식, 외부의 리소스 사용 여부 등등에 따라 다양한 접근방법이 사용되고 있다. 하지만 이들을 그들이 사용한 검색의 대상에 따라 구분을 하면 크게 문서 검색(document retrieval)과 단락 검색(passage retrieval)으로 분류해 볼 수 있다.

문서 검색을 수행하는 대표적인 시스템들로는 [2], [3], [4] 등이 있는데, 이들의 공통적인 특징은 검색된 문서들에 대해서 자신들만의 독특한 필터링(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]들이 있다. [5]는 전처리 작업을 통하여 문서들을 일정 크기의 단락으로 미리 나누어 놓은 후, 이를 색인 하여 단락 검색을 수행한다. 이와 달리 [6]에서는 문서의 임의의 범위를 대상으로 스코어를 계산하여 검색한 후 그 중앙을 기준으로 고정된 크기의 단락을 뽑아낸다. 이러한 단락 검색 방식들은 문서 검색 방법에 비해 검색시 계산량이 조금 더 많긴 하지만, 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 질의 응답 시스템의 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 의존적이 될 수 밖에 없다. 이렇듯 질의 응답 시스템에서 사용되는 검색 시스템은 질의 응답 시스템의 상한 성능을 제한하는 중요한 요소이다.

그러나 기존 연구에서는 검색 시스템 자체에 대한 성능평가가 이루어지지 않았다. 즉, 해당 검색 시스템은 정답을 얼마나 잘 찾아내고 있는지, 정답을 찾기 위해서 얼마나 많은 양을 검색해 내고 있는지, 해당 검색 시스템에서 채용한 여러 기법들이 이러한 결과에 어떻게 기여를 했는지 등에 관한 분석이 전무한 실정이다.

이에 본 논문에서는 질의 응답 시스템을 위한 검색 방법 중 단락 검색 방법의 성능을 분석해 보고자 한다. 특히 단락 검색의 성능에 영향을 미치는 요인은 여러 가지가 있겠지만, 단락의 정의 방식에 따라 단락 검색이 어떠한 성능의 차이를 보이는가 하는 점에 초점을 맞추었다.

[1] 이후 검색대상 텍스트 영역을 단락이라 칭하기로 한다.

3. 단락 검색(passage retrieval)

단락 검색(passage retrieval)은 문서 전체를 대상으로 검색을 수행하지 않고, 문서의 일부분만을 대상으로 검색을 수행한다. 여기에서 단락(passage)이란 검색 대상이 되는 문서의 일부분을 지칭하는데, 단락으로 정의될 수 있는 가장 큰 단위는 문서이고, 가장 작은 단위는 단어이다. 그리고 문서와 단어 사이의 크기를 갖는 단락들이 다양하게 정의될 수 있다. 가령 하나 혹은 n개의 문장을 하나의 단락으로 정의할 수 있고, n개의 단어를 하나의 단락으로 정의할 수도 있으며, 또한 임의의 길이의 단락 역시 정의될 수 있다.

최근의 질의 응답에 관련된 연구 동향을 살펴보면, 성능 향상을 위하여 구문 분석, 의미 분석 등 복잡도가 높은 다양한 자연어 처리 기법들을 점점 더 많이 사용하고 있는 추세이고[2][7], 이러한 점을 고려해 볼 때 질의 응답 시스템에서의 정보 검색 단계는 정답 추출 단계로 전달해 줄 단락의 기본 단위를 문장이 되도록 하는 것이 적합하다.

이를 바탕으로 본 논문에서는 고정된 n개의 문장을 하나의 단락으로 정의하는 고정 길이 단락 검색(fixed length passage retrieval)과 임의의 개수의 문장을 하나의 단락으로 정의하는 가변 길이 단락 검색(variable length passage retrieval) 두 가지 방법에 관해 고찰해보고자 한다.

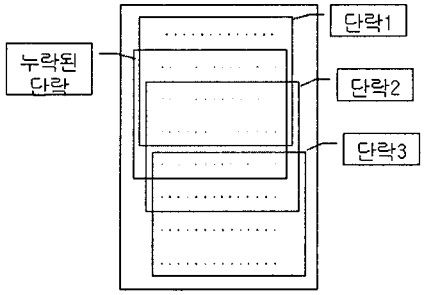
3.1 고정 길이 단락 검색

3.1.1 단락의 구성

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

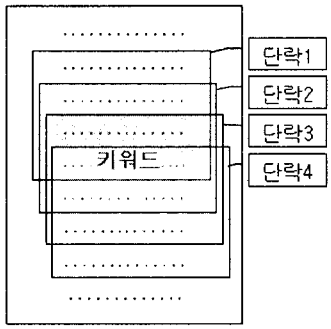
여기에서 단락의 크기에 대한 정의 이외에도 고정 길이 단락 검색을 수행함에 있어 고려되어야 할 또 다른 사항이 있는데, 바로 단락이 결정되는 시점이다. 단락을 결정하는 시점은 색인을 하는 시점과, 검색을 하는 시점 두 가지 경우가 있다.

색인시 단락을 결정하는 경우엔 일반적으로 단락의 반이 다른 단락과 중첩되도록 문서에서 단락들을 추출하여 색인한 뒤, 이러한 단락들을 독립된 작은 문서로 간주하여 검색을 진행한다[8]. 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 색인시 단락을 결정할 경우 색인에서 누락되는 단락이 발생하는 모습을 보여준다.



[그림 1] 색인된 단락 및 색인에서 누락된 단락들

검색시 단락을 결정하는 경우엔 키워드들의 출현 위치 정보를 색인시점에 미리 저장한 뒤 검색시 문서 내에서 키워드가 출현하는 위치를 기점으로 단락을 추출한다. 즉, 키워드가 출현한 문장을 포함하고 있는 연속된 n개의 문장을 단락으로 추출한다. 이 방식은 색인시 단락을 결정하는 방법에 비해 계산량이 늘어나는 단점이 있지만, 존재할 수 있는 모든 단락들을 고려할 수 있다는 장점이 있다. [그림 2]는 검색시 단락을 결정하는 모습을 보여준다.



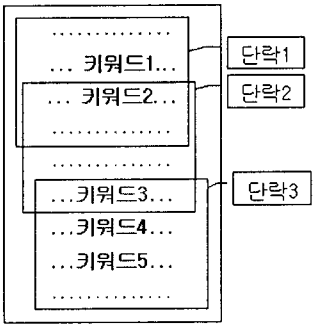
[그림 2] 검색시점에 추출되는 단락들의 모습

본 논문에서는 단락을 결정하는 시점으로 검색시점을 선택하여 비록 계산량이 늘어나지만 단락이 생성될 수 있는 가능한 모든 경우를 고려할 수 있도록 하였다.

3.1.2 중첩된 단락 처리

검색 시점에 단락을 결정하는 경우에는 비슷한 내용을 포함하고 있는 중첩된 단락들이 다수개가 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 일으키게 된다. 이러한 단점을 보완하기 위해, 검색시 키워드가 출현한 각각의 문장에 대해서 해당 문장을 기준으로 추출 가능한 단락들 중에 스코어가 가장 높은 단락 하나만을 추출하도록 제한을 가한다. [그림 3]은 중첩된 단락을 처리하고 난 후 최종적으로 추출되는 단락들의 모습을 일례로 보여주고 있다.



[그림 3] 중첩된 단락을 처리한 후의 모습

3.1.3 스코어 계산 방법

각 단락의 스코어 계산은 TREC에서 이미 그 성능이 검증된 Okapi 시스템의 BM25를 단락 검색에 적합하도록 파라미터들의 의미를 약간 수정한 [식 1][2]을 사용하였다[9].

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

$$\text{where, } K = k_1((1-b)+b \cdot pl/avpl),$$

$$k_1 = 1.2,\ b = 0.75,\ k_3 = 1000$$

[식 1] 스코어 계산식

본 논문에서는 색인시점에 모든 단락들을 구성하여 색인하지 않고 검색시에 동적으로 단락을 구성하기 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. 그러나 모든 단락의 크기가 m 문장임을 고려하면, N'과 n'은 각각 S/m, s/m 으로 그 값을 근사하여 사용할 수 있다.[3] 이때, S/m은 컬렉션에서 중첩된 단락들을 제거하였을 경우 생성될 수 있는 모든 단락들의 개수에 대한 근사값을, s/m은 중첩되지 않게 뽑혀진 단락들 중 해당 키워드를 포함한 단락들의 개수에 대한 근사값을 각각 나타낸다.

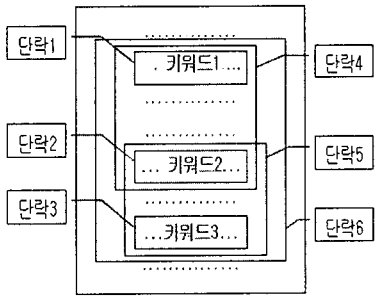
최적의 검색 성능을 얻기 위해서는 N' 및 n'에 대해서 좀 더 정확한 근사값을 사용하고, 여러 실험을 통해 k1, b, k3 값을 설정해야 하겠지만, 본 논문에서는 단락의 특성에 따른 검색 성능의 추이를 살피는 것이 목적이므로 이러한 파라미터들에 대해서는 단순히 앞서 설명한 값들을 그대로 사용하기로 한다.

3.2 가변 길이 단락 검색

3.2.1 단락의 구성

가변 길이 단락 검색에서는 단락의 크기가 임의의 개수의 문장으로 정의된다. 즉, 문서 전체가 하나의 단락이 될 수도 있고, 문서의 네번째 문장 하나가 단락이 될 수 있으며, 문서의 일곱번째 문장부터 열번째 문장까지가 하나의 단락이 될 수도 있다. 따라서 색인시점에 단락을 결정할 수는 없고, 검색시점에서만 단락이 결정된다.

임의의 크기로 임의의 위치에서 단락이 구성될 수 있기 때문에, 가능한 모든 단락을 대상으로 스코어를 계산하고 검색을 수행하는 것은 현실적으로 불가능 하다. 그러나 스코어를 계산할 때 사용이 되는 BM25의 특성을 이용하면 이 문제를 해결할 수 있다.



[그림 4] 가변 길이 단락의 구성

두 개의 단락 내에 키워드들이 동일하게 각각 포함되어 있을 경우 BM25는 보다 더 짧은 단락을 선호하는 특성이 있다. 즉, 동일한 키워드들을 포함

[2] [식 1]에서 N'은 컬렉션 내 전체 단락들의 수, n'은 해당 키워드를 포함하고 있는 단락들의 수, tf'는 특정 단락내에서 키워드의 출현 빈도수, qtf는 질의에서 키워드의 출현 빈도수, pl은 단락의 길이, avpl은 평균 단락 길이를 각각 의미한다.

[3] S는 컬렉션 내 전체 문장들의 수, s'은 해당 키워드를 포함하고 있는 문장들의 수를 각각 의미한다

하는 여러 단락들 중에서는 가장 짧은 단락, 다시 말해 키워드를 포함하고 있는 맨 처음과 마지막 문장을 경계로 하는 단락이 항상 선호된다. 따라서 키워드를 포함하고 있는 문장들이 단락의 시작 및 끝의 경계선이 되는 단락들만을 유효한 단락으로 고려하면 된다.

3.2.2 중첩된 단락 처리

가변 길이 단락 검색에서는 검색된 각 단락들이 그 크기에 따라 각기 다양한 양상으로 다른 단락들과 중첩되어 있다. 또한 문서에 출현한 모든 키워드들을 포함하는 단락, 즉, 가장 긴 단락은 다른 모든 단락들과 중첩되어 있다. 그러므로 가변 길이 단락 검색을 하면 모든 단락은 적어도 하나 이상의 중첩된 단락이 존재하게 된다. 결국 중첩된 단락 중 스코어가 가장 높은 단락 하나만을 추출하게 되면 한 문서에서 하나의 단락만이 추출된다.

3.2.3 스코어 계산 방법

가변 길이 단락 검색 역시 고정 길이 단락 검색과 동일하게 [식 1]을 사용하였다. 하지만 고정 길이 단락 검색과는 달리 단락의 크기가 일정치 않은 가변 길이 단락 검색에서는 평균 단락 길이(avpl)를 근사할 방법이 없다. 어쩔 수 없이, 테스트 컬렉션의 문서당 평균 문장수가 대략 22문장임을 고려할 때, 평균 단락 길이 avpl에 대해서는 1문장부터 22문장까지 그 값을 변화해가며 실험을 수행하였다. 그리고 고정 길이 단락 검색과 유사하게 N'은 S/avpl, n'은 s/avpl로 각각 그 값을 근사하여 사용하였다.

4. 실험 및 평가

4.1 실험 환경

본 논문에서 구현한 시스템은 TREC Q&A track 테스트 컬렉션에 적용하여 평가하였다. Q&A track 컬렉션은 대략 100만 건의 각종 신문기사로 이루어져있다. 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 중 1 ~ 100번 까지 총 100개의 질의들을 사용하였다.

4.2 평가 방법

실험 결과의 평가 방법은 전체 질의 중 제한된 검색량 내에서 정답을 찾은 질의의 개수, 즉, [식 2]와 같은 재현율을 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 순위화 하여 배열한 뒤 상위 n개의 문장을 최종적인 검색 결과로 추출하여 그 안에 정답이 포함되어 있는지를 평가하였다.

$$\text{재현율} = \frac{\text{정답을 찾은 질의의 수}}{\text{전체 질의의 수}} \times 100$$

[식 2] 재현율

4.3 실험 결과 및 분석

4.3.1 고정 길이 단락 검색

고정 길이 단락 검색에 대해서는 테스트 컬렉션의 문서당 평균 문장수가 대략 22문장임을 고려할 때, 단락의 크기를 1문장에서부터 22문장까지 총 22가지의 단락에 대해 각각 실험하였고, 최고 1000 문장까지의 검색량에 대해 실험을 하였다.

실험 결과는 [표 1]과 같다. [표 1]의 상변에서 p1 ~ p22는 단락의 크기를 1~ 22문장으로 설정하여 검색한 결과를, doc은 문서 검색을 수행한 결과를 각기 나타낸다. 표의 좌변에 있는 숫자가 의미하는 것은 검색된 문장의 양이다. 즉, '100'은 상위

[표 1] 고정 길이 단락 검색 및 문서 검색 결과

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p12	p15	p17	p22	doc
100	90	89	89		87	83	82	80	79	84	81	80	82	80	85
200	91	92			92	92	89	87	88	87	88	89	87	88	
300	92	93			94	94	92	90	90	91	94	94	92	89	93
400	94	94			94			93	93	93				94	94
500	95				95	96	95	95	94	94				95	95
600	95	96		96	95	96		95	95	94	96	96	96	96	95
700	95	96			95			96	95	95	96	96	96	96	95
800	96	96			96			97	95	95	96	96	96	96	95
900		96			96			97			96	96	96	96	96
1000		96		97	97	97	97	97	97	97	96	96	96	96	96

100개까지의 문장들을, '500'은 상위 500개까지의 문장들을 검색한 결과를 나타낸다. 그리고 표의 내용부분의 숫자가 의미하는 것은 재현율이다. 예를 들어, 상변이 p4이고 좌변이 500인 경우엔 96의 값을 갖는데, 이것은 단락의 크기를 4문장으로 고정하고 검색을 수행하여 상위 500개의 문장을 결과로 얻어내었을 때, 96개의 질의에 대해서는 검색결과 내에 정답이 포함되어 있고 나머지 4개의 질의에 대해서는 정답을 찾지 못했음을 나타낸다.

[표 1]의 실험 결과를 살펴보면 문서 검색 보다는 고정 길이 단락 검색이 보다 더 우수한 성능을 나타내고 있다. 그리고 단락의 크기가 큰 경우보다는 작을 경우 대체로 성능이 우수하고, 단락의 크기가 3 혹은 4문장일 경우 가장 좋은 성능을 나타낸다. 문서의 특정 부분에 정답이 있는지를 확인하기 위해서는 그 주변의 3, 4 문장을 살펴보는 것만으로도 정답의 유무를 대체로 확인해 낼 수 있음을 본 실험 결과에서 유추해 낼 수 있다.

전체적으로 단락의 크기가 큰 경우보다 작은 경우에 비교적 검색 성능이 우수한 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.

4.3.2 가변 길이 단락 검색

가변 길이 단락 검색에서는 평균 단락 길이를 측정할 방법이 없으므로 3.2.3절에서 언급한대로 BM25의 평균 단락 길이 avpl 파라미터를 1문장에서 22문장으로 각각 변화해가며 실험하였고, 그 결과는 [표 2]와 같다. 표의 상변에 있는 ap1 ~ ap22는 평균 단락 길이 avpl을 1 ~ 22 문장으로 각각 설정했음을 의미하고, 좌변은 검색된 문장의 양을 나타낸다. 표의 맨 마지막 줄에 있는 avpl'은 상위 1000개의 단락을 검색했을 때 실제로 검색된 단락들의 평균 단락 길이이다. 즉, 테스트 컬렉션 전체에서의 평균 단락 길이를 표의 상변에 있는 값으로 설정했을 경우 검색된 단락들의 실제 평균 단락 길이가 표의 맨 마지막 줄에 나타나 있다.

[표 2]의 가변 길이 단락 검색 실험 결과를 살펴보면 검색을 할 때 미리 설정하는 평균 단락 길이 avpl에 비해 실제로 검색된 단락들의 평균 길이가 훨씬 더 짧다는 특징이 우선 눈에 들어온다. 즉, 가변 길이 단락 검색은 크기가 작은 단락을 선호하는 특성이 있음을 알 수 있다.

검색 성능은 고정 길이 단락 검색 결과와 비교했을 때 제한된 검색량이 크고 작음에 상관없이 전체적으로 보다 우수한 성능을 나타낸다. 특히, 고정 길이 단락 검색에서의 가장 좋은 결과가 상위 1000위에서 조차도 98%의 재현율을 보이는데 반해, 평균 단락 길이(avpl)가 7로 설정된 가변 길이 단락 검색의 경우 상위 900문장만으로도 100%의 재현율을 보인다.

이와 같이 가변 길이 단락 검색이 고정 길이 단락 검색에 비해 성능이 우수하게 나온 이유는 고정 길이 단락 검색에서 잘 찾아내지 못하거나 검색에 실패하는 질의들에 대해 가변 길이 단락 검색 방법이 검색을 잘 해내기 때문이다.

[표 1]을 살펴보면, 고정 길이 단락 검색에서는 단락의 크기에 따라 그리고 검색한 양에 따라 다소 정도의 차이가 있긴 하지만 검색에 실패한 질의들이 몇몇씩 항상 존재하고 있음을 알 수 있다. 그러나, 검색에 실패한 질의들의 개별적 실험 결과를 분석해 보았을 때, 단락의 크기에 따른 모든 경우에 대해 검색에 실패한 질의는 전무하였다. 즉, 단락의 크기가 큰 경우 검색에 실패한 질의는 단락의 크기가 작은 경우 검색에 성공을 하고, 단락의 크기가 작은 경우 검색에 실패한 질의는 단락이 큰 경우 검색에 성공하였다. 이러한 사실로 미루어 보아 고정 길이 단락 검색의 경우 검색에 실패하는

[표 2] 가변 길이 단락 검색 결과

	ap1	ap2	ap3	ap4	ap5	ap6	ap7	ap8	ap9	ap10	ap12	ap15	ap17	ap22
100	89	91				92	91	91	91	90	91	92	91	91
200	90	93					95	94	93	93	93	95	95	95
300	91	95		96	96	96	96	96	94	94	94	96	96	96
400	92	96		96	96	96	96	96	95	95	95	96	96	96
500	92	96				96	96	96	96	96	96	96	96	96
600	93	96	97	97		97	96	96	97	96	96	96	96	96
700	94						97	96	97	97	96	97	96	96
800	95	98	98	98	98		98	98	98	98	96	97	96	96
900	95	98	98	99	99	99		99	99	98	96	97	97	96
1000	95	98	98	99	99				99	99	98	97	97	96
avpl'	1.2	1.4	1.7	2	2.3	2.6	3	3.3	3.6	3.9	4.6	5.5	6.1	7.4

요인이 단락의 크기에 따라 서로 다를 것이라는 점을 알 수 있다.

[표 3]은 검색이 잘 되지 않는 일부 질의에 대해 고정 길이 단락 검색 및 가변 길이 단락 검색의 검색 결과를 보여준다. 표에서 상변은 각각의 질의 번호를 나타내고, 좌변에서 p1 ~ p22는 고정 길이 단락 검색 결과를, ap7은 평균 단락 길이 avpl을 7로 설정한 가변 길이 단락 검색 결과를 각각 나타낸다. 그리고 표의 내용 부분의 숫자는 정답이 검색된 순위를 의미하며, n은 1000위까지의 검색결과에서도 정답을 찾지 못했음을 의미한다.

[표 3] 일부 질의에 대한 개별적 검색 결과

	43	96	9	98	97
p1	719	967	45	13	832
p2	16	449	67	258	n
p3	3	153	229	950	n
p4	1	99	244	n	n
p5	2	98	278	n	n
p6	2	15	439	n	n
p7	1	31	573	n	n
p8	1	59	701	n	n
p9	1	57	862	n	n
p10	1	63	849	n	n
p12	2	221	N	n	n
p15	1	118	N	n	n
p17	1	98	N	n	n
p22	1	443	N	n	n
ap7	3	35	63	205	673

먼저, 98번 질의의 경우 단락의 크기가 4문장 이상일 경우 모두 검색에 실패했다. 그러나 단락의 크기가 3문장 이하인 경우, 특히 1문장인 경우엔 검색 결과가 상당히 좋다. 9번 질의 역시 비슷한 양상으로 단락의 크기가 커짐에 따라 검색 성능이 저하되는 모습을 보이고 있다. 이러한 질의들에 대해서 실제 검색된 단락들을 살펴보면, 정답이 출현한 문장 안에는 질의에서 사용된 키워드들이 적당히 존재하고 있지만, 정답이 출현하지 않은 다른 단락들에는 더욱 더 많은 키워드들이 존재하고 있음을 확인할 수 있다. 고정 길이 단락의 크기를 크게 설정하여 검색할 경우 한 문장 안에 적당한 양의 키워드들이 들어있는 것보다는 단락 안에 키워드들이 다수 포함되어 있는 것을 선호하게 되고 그 결과 실제 정답이 출현한 단락의 순위는 하락하게 된다. 하지만 이러한 질의에 대한 가변 길이 단락 검색의 성능은 괜찮은 편이다. 이는 가변 길이 단락 검색이 크기가 큰 단락보다는 작은 단락을 선호하는 특성이 있기 때문이다.

43번 질의의 검색 결과를 살펴보면 단락의 크기가 2 이상인 경우엔 정답이 아주 높은 순위로 검색되는 반면, 단락의 크기가 1인 경우 순위가 상당히 낮게 내려간다. 이 43번 질의에 대해서 실제 검색된 단락들을 살펴보면, 정답이 포함된 문장 안에는 키워드들이 별로 없는 반면에 바로 앞 문장에는 키워드들이 많이 모여있음을 확인할 수 있다. 이런 이유로 단락의 크기가 1인 경우 앞 문장의 키워드들을 고려할 수 없게 되고 결국 크기가 2 이상인 단락들에 비해 성능이 떨어질 수 밖에 없다. 96번 질의 역시 비슷한 경우로서, 정답이 포함된 문장엔 키워드들이 별로 없고 정답 주변의 여섯 문장에 걸쳐 키워드들이 분산되어 분포하고 있다. 따라서 단락의 크기가 작은 경우 보다는 약간 큰 경우에 좋은 성능을 보이게 된다. 43, 96번 질의에 대한 가변 길이 단락 검색의 결과를 보면 두 개의 질의에 대해 모두 좋은 성능을 나타내고 있는데, 이는 가변 길이 단락 검색이 작은 단락을 선호하는 경향이 있음에도 불구하고 작은 단락이 그리 유용하지 않다면 적절한 크기의 단락을 선정해내는 능력이 있음을 보여준다.

97번 질의의 경우 단락의 크기에 상관없이 대부분 좋지 못한 검색 결과를 보인다. 이러한 종류의 결과를 보이는 질의들은 키워드들이 동의어나 다른 표현들로 변환되어서 정답 주위에 출현하는 특징이 있다. 이렇게 질의에 사용된 키워드 자체가 변환되어 정답 주위에 출현하는 질의들에 대해서는 가변 길이 단락 검색 역시 그리 좋은 성능을 나타내지는 못하고 있다.

지금까지 살펴본 분석 결과를 종합해보면, 가변 길이 단락 검색은 작은 크기의 단락 검색이 갖는 특성과 큰 크기의 단락 검색이 갖는 특성의 장점들이 적절히 결합되어 결국 고정 길이의 단락 검색보다 우수한 성능을 갖게 됨을 알 수 있다.

5. 결론

본 논문에서는 질의 응답 시스템을 위한 단락 검색 시스템에서 단락의 크기가 검색에 미치는 영향을 분석하였고, 고정 길이 및 가변 길이 단락을 사용하는 단락 검색 방법을 제안하였다.

고정 길이 단락 검색의 경우 단락의 크기가 3 ~ 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.

가변 길이 단락 검색의 경우 작은 크기의 단락 검색이 갖는 특성과 큰 크기의 단락 검색이 갖는 특성의 장점들이 적절히 결합되기 때문에 결국 고정 길이의 단락 검색보다 우수한 성능을 갖게 됨을 밝혀내었다.

본 논문에서는 단락 검색 시스템의 검색 성능을 향상시키기 위한 한가지 접근 방법으로서 단락을 정의하는 방법에 초점을 맞추었는데, 질의의 구성

방식이나 다양한 스코어 계산 방식, 적합성 피드백 기법 등을 접목시키면 아직도 성능이 개선될 여지가 많이 남아있다.

6. 참고 문헌

[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] A. Ittyscheriah, M. Franz, S. Roukos, "*IBM's Statistical Question Answering System*", In the Proceedings of Text REtrieval Conference (TREC 2001), 2001.

[8] 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.

[9] 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.