질의 응답 시스템에서 구문 근접성에 기반한 정답 후보 랭킹 방법

나승훈[0] 강인수 권오욱 이종혁
포항공대 컴퓨터 공학과
(nsh[0], dbaisk, ohwoog, jhlee)@postech.ac.kr

Answer Candidate Ranking based on Syntactic Proximity in Question Answering

Seung-Hoon Na[0] In-Su Kang Oh-Woog Kwon Jong-Hyeok Lee
Div. of Electrical and Computer Engineering
Pohang University of Science and Technology

요 약

질의 응답 시스템의 성능을 높이기 위해서는 정답 후보(Answer Candidate)를 랭킹하는 방법이 매우 중요하다. 본 논문에서는 기존의 정답 후보 랭킹을 위해 사용하던 위치 근접성의 문제점을 제시하고, 이를 보완하기 위한 구문 근접성을 이용하는 방법에 대해 제안한다. 실험 결과는 논문에서 제안한 구문 근접성을 사용한 정답 후보 랭킹 방법이 위치 근접성을 이용한 방법보다 더 개선된 방법임을 보여준다.

1. 서 론

사용자가 문서가 아닌 답을 원할 때 그 사용자가 기존의 문서 검색을 이용하여 원하는 목적을 달성하기는 어렵다. 질의 응답 시스템(Question Answering System: QA)은 이러한 어려움을 해결하기 위해, 대용량의 데이터 모음으로부터 사용자의 다양한 자연어 질문을 입력으로 받아 문서가 아닌 정답을 제공해주는 시스템이다[11].

일반적인 QA시스템의 구조는 다음과 같이 6가지의 컴포넌트로 나누어 기술될 수 있다[6].

1)질문 분석(Question Analysis), 2) 문서 집합 전처리(Document Collection Preprocessing), 3) 후보 문서 선택(Candidate Document Selection), 4) 후보 문서 분석(Candidate Document Analysis), 5)정답 추출(Answer Extraction), 6) 응답 생성(Response Generation)

시스템의 흐름은 2)를 제외하고 번호가 작은 순서부터 진행되게 되는데, QA시스템들은 5)의 정답 추출에서 가장 많은 에러가 있었다고 보고하였다. 다시 말해, 3)에서 선택된 상위의 단락들 내에서는 정답이 존재하였고 4)에서 정답 후보의 인식률도 그리 낮지 않았지만, 5)의 결과로 나온 상위 5개의 결과에는 답이 존재하지 않는 경우가 많다는 것이다[8].

기존의 QA시스템들은 정답 후보의 개수를 줄이거나, 정답 랭킹 방법을 개선하는 등 다각도로 정답 추출의 어려움을 해결하려고 하였다. [7]은 질문과 정답 후보가 위치한 문맥 텍스트를 논리형태로 변환하여, 정답 추출을 문맥의 논리형태와 질문의 논리형태 간의 추론문제로 바꿈으로써 정답 후보의 개수를 줄이려고 시도하였다. [4]에서는 정답 후보가 여러번 중복될수록 그 정답 후보가 답일 확률을 높여 정답 랭킹 방법을 개선하였다. 이 밖에, [3]에서는 정답 유형의 사용만으로 높은 성능 향상을 보였으며, [9]은 질문과 단락을 단일화 문법(Unification-based Grammar)을 사용하여 분석하여 매칭하는 방법을 제안하였다.

이러한 분위기 속에서 출발한 본 연구는 보다 나은 정답 추출을 위해, 기존의 QA 시스템들이 사용하고 있는 단어 간의 근접성의 문제점에 대해서 고찰하고, 그것을 보다 개선한 구문 근접성의 사용을 제안하고자 한다.

2. 근접성 기반 정답 후보 랭킹

정답 후보 랭킹은, 정답 후보가 나타난 단락의 내용이 질문의 내용을 얼마나 잘 반영하였는가를 계산하는 문제라고 할 수 있다. 단락이 질문의 내용을 많이 반영하면 할수록 그 단락은 질문과 관련이 높게 되고, 결국 단락 내의 정답 후보가 실제 정답일 확률은 커지게 될 것이다.

정답 후보 랭킹을 위해 사용되는 특질(feature)은 여러 가지가 있다. 그 중에서도 근접성(Proximity)은 문장에서 두 단어간의 의미적 관계를 형태적인 거리로 해석한 것으로써 많은 QA시스템에서 사용하는 기본적인 특질중의 하나이다.

근접성을 사용한 모든 가능한 정답 후보 랭킹 방법들은 질문과 단락을 그래프로 해석하여 질문의 그래프와 단락의 그래프의 유사도를 계산하는 문제로 볼 수 있다. 그러므로, 근접성을 사용한 방법론은 문장을 그래프로 해석할 때 취하는 토폴로지(topology)에 따라 성능이 좌우된다.

일반적으로 근접성에 기반한 정답 후보 랭킹은 아래의 <정의 1>과 같다.

<정의1>.질문 그래프와 단락 그래프를 각각 $G(V,E)$ 과 $G'(V',E')$ 라고 하자. 단락 그래프 G' 의 주어진 정답 후보가 $t_k \in V'$ 이고, 질문 그래프에서 $N-1$ 개의 노드 $q_i \in V$ (**E**, $i \neq k$, $1 \le i \le N$)가 단락의 그래프에서 $t_i \in V'$로 나타났다고 할 때, 정답 후보 x의 점수는 아래와 같이 정의된다.

$$score(x) = \max_{t_1,\cdots,t_N} \left\{ \sum_{(q_i,q_j)\in E} \frac{w_{ij}}{d(t_i,t_j)} \right\}$$

여기서, $d(t_i,t_j)$는 단락의 그래프 G'상에서 두 노드 t_i,t_j 간의 경로의 비용을 의미하고, w_{ij}는 에지(edge) (q_i,q_j)의 중요도를 의미한다.

<정의 1>에 따르면 정답 후보의 점수는, 질문 그래프의 각 에지에 참여하는 단어들이 단락 그래프상에서 더 근접하여 나타날수록 높은 값을 갖게 된다. 다시 말해, 질문의 그래프상의 에지가 단락 그래프 상에서 형태적으로 보존되면 될수록 단락의 내용은 질문의 내용과 유사하다는 것이다.

결국 <정의 1>에 따르면, 문장 그래프의 토폴로지가 근접성 기반 정답 후보 랭킹에 매우 중요한 역할을 한다는 것을 알 수 있다. 본 연구에서는 기존의 방법들이 가정했던 문장 그래프의 형태에 대한 문제점을 고찰하고, 새로운 그래프의 토폴로지를 제안하고자 한다.

3. 위치 근접성 (Positional-Proximity)

위치 근접성은 두 단어간의 어휘적 거리를 의미하는 것으로써, 그래프를 구성하는 에지가 단락상에서 순서상 인접해 있는 두 단어로부터 형성된다.

즉, 문장이 순서대로 $s_1,\cdots,s_i,\cdots,s_M$ 로 이루어졌다고 하면, 위치 근접성에서의 문장 그래프의 토폴로지는 아래의 <그림 1>과 같다.

$$s_1 \text{ —— } \cdots \text{ —— } s_i \text{ —— } \cdots \text{ —— } s_M$$

<그림 1> 위치 근접성에서 가정하는 문장 그래프의 토폴로지

위치 근접성은 문장을 위치상의 거리만을 가지고 점수를 계산하기 때문에 빠르다는 장점이 있지만, 아래의 <예 1>에서와 같이 정답이 아닌 정답 후보에 높은 점수를 부여할 위험이 크다.

<예 1>

질문: Who invented the paper clip?

단락1: " Like the guy who *invented* the safety pin, or the guy who *invented the paper clip*" , *David* says.

단락2: The *paper clip*, weighing a desk-crushing 1,320 pounds, is faithful copy of Norwegian, *Johan Vaaler*' s 1899 *invention*, said *Per Langker* of the Norwegian School of Management

<예 1>에서 질문의 단어인 *invented*, *paper*, *clip*이 단락1과 단락2에 나타난 부분이 이태릭체로 표시되어 있으며, 정답 후보인 경우에는 밑줄이 추가되었다. 만약 단락1의 *invented*와 같이 동일한 질문상의 단어가 단락에 여러 번 나타나는 경우에는 정답 후보와 가장 가까운 단어만을 취하고 나머지는 버리게 된다. 만약 위치 근접성을 사용하여 정답 후보를 랭킹 한다면 실제의 답인 *Johan Vaaler*보다 *David*가 더 높은 점수를 얻게 된다.

4. 구문 근접성(Syntactic-Proximity)

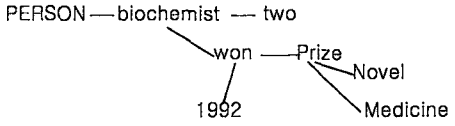
<예 1>과 같은 오류를 줄이기 위해서는 단락을 보다 정교한 자연어 처리를 사용하여 분석하고 그 결과를 정답 후보의 랭킹에 반영해야 한다. 그렇다고 질문에서 나타난 구문 관계와 단락상에 나타난 구문 관계를 정확하게 매칭하려고 시도하는 것은 , 수많은 말바꿈(paraphrasing) 현상이 가능하기 때문에, 정답의 재현률의 감소를 초래할 수 있다.

그러므로, 우리는 구문 관계의 사용을 피하고, 구문 트리의 형태만을 사용하여, 위치 근접성에서 가정했던 단순한 토폴로지를 개선시키려고 한다. 구문 근접성은 구문 그래프상에서의 두 노드 간의 거리를 의미하는 것으로, 구문 관계를 전혀 고려하지 않기 때문에 재현율의 감소를 초래하지 않으며, 앞 장에서의 위치 근접성의 정확률을 개선시킬 수 있다.

구문 근접성을 사용하기 위해, 먼저 질문과 단락을 구문 분석기를 통해 구문 트리를 얻어 내고, 이것을 다시 의존 트리로 변환시킨다[7]. 이렇게 생성된 의존 트리의 각 에지마다, 그 에지의 중요도 w_{ij}를 계산한다. 여기서 에지의 중요도는 그 에지에 참여하는 두 단어의 간의 구문관계나, 어휘관계, 그리고 단어의 가중치에 따라 결정된다.

아래의 <예 2>를 의존 트리로 바꾸게 되면 <그림 2>과 같이 나타낼 수 있다.

<예 2>" Who two US biochemists won the Nobel Prize in Medicine in 1992?"



<그림 2>: <예 2>에 대한 의존 트리

5. 실험 및 결과

우리는 TREC8의 QA track의 질문 중 정답 유형이 사람에 해당하는 질문 53개를 추출하여 논문에서 제안한 구문 근접성의 유용성을 입증하기 위해 세 가지의 랭킹 방법을 비교 실험하였다

첫 번째 랭킹 방법은, 랜덤 근접성을 사용한 랭킹 방법으로써 질문을 랜덤 트리로 해석하는 방법이다.

두 번째 랭킹 방법은, 3장에서 언급한 위치 근접성에 기반한 랭킹 방법으로, 질문과 단락을 1차원적인 그래프 형태로 해석하는 방법이다

세 번째 랭킹 방법은, 4장에서 언급한 구문 근접성에 기반한

랭킹 방법으로, 문장을 구문 그래프(트리)로 해석하는 방법이다. 실험에서는 질문에 대해서만 구문 근접성을 고려하였고 단락에 대해서는 위치 근접성을 사용하였다.

모든 랭킹 방법에서, 문서 검색 시스템은 벡터 모델에 기반한 자체 검색 엔진을 사용하였으며, 이 검색 엔진이 제공하는 상위 500개의 문서로부터 <정의 1>에 따라 계산된 점수의 값이 가장 높은 상위 5개의 250bytes 길이의 passage를 답으로 추출하게 된다. 모든 실험에서 에지의 중요도는 $w_{ij}=1$로 동일하다고 가정하였다.

이렇게 하여 실험한 결과가 <표 1>에 나와 있다.

<표 1> 실험 결과

정답 후보 랭킹 방법	*MRR*	*TOP1*
랜덤 근접성	0.439322	0.320755
위치 근접성	0.47334	0.339623
구문 근접성	0.54535	0.433962

<표 1>에서 세 번째 열인 TOP1은 전체 질문 중 상위 1위로 랭킹된 질문의 비율을 가리킨다.

실험 결과는 위치 근접성을 사용한 방법보다 구문 근접성을 사용한 방법이 MRR로 평가 했을 때 약 13%의 성능 향상을 보여주었다.

단락에서도 구문 근접성을 이용하였다면 구문 근접성을 정답 후보 랭킹 방법의 성능은 더욱 향상되었을 것이다.

6. 결론

본 논문에서는 질의 응답 시스템에서의 정답 후보 랭킹 방법의 개선의 필요성에 대해 인식하고, 정답 후보 랭킹에 유용하게 사용되는 특질인 근접성에 대해 고찰해 보았다.

우리는 문장 그래프의 토폴로지가 근접성에 기반한 정답 후보 랭킹 방법의 성능에 많은 영향을 끼칠 것이라 보고, 기존의 1차원적인 그래프의 토폴로지를 구문 정보를 고려한 토폴로지로 개선시킨 구문 근접성을 사용하는 방법을 제안하였다.

여기서는 구문 그래프상의 에지의 비용이 모두 동일하다고 간주하였으나, 구문 그래프의 에지에 비용을 적절히 부여한다면 여기서 언급한 방법을 좀 더 개선시킬 수 있을것이다.

그러나 본 논문의 방법은 구문 분석의 사용의 장점을 언급한 것이지, 구문 분석의 결과를 최대한 이용한 것은 아니다. 예를 들어, 본 논문에서 다루었던 근접성 대신 구문 관계의 보존성을 사용하여 정답 후보의 랭킹을 사용할 수 있다. 그러나 아직까지는,이러한 구문 관계의 보존성은 적용하기 힘들다. 구문 관계의 보존성을 적용하기 위해서는 다양한 말바꿈이나 생략 현상을 인식하기 위한 지식의 구축이 필수적이기 때문이다. 앞으로의 QA시스템의 정답 후보 랭킹 방법은 구문 관계 보존성 혹은 의미 관계 보존성등을 적용하는 방향으로 나아가야 하며, 그러기 위해 정교한 자연어 처리 기법의 활용과 충분한 지식 체계의 구축이 연구되어야 할 것이다.

감사의 글

본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

참고 문헌

[1] S. Abney, M. Collins, A. Singhal, " Answer Extraction" , In Sixth Applied Natural Language Processing Conference, 2000

[2] J. Burger, " Issues, Tasks, Program Structures to Roadmap Research in Question & Answering(Q&A)" , NIST, 2001

[3] C. Cardie, " Examining the Role of Statistical and Linguistic Knowledge Sources in a General-Knowledge Question-Answering System" , In Sixth Applied Natural Language Processing Conference, pp.180-187, 2000

[4] C.L.A Clarke, G.V. Cormark, T.R Lynam, " Exploiting Redundancy in Question Answering" , In 24rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.358-365, 2001

[5] G.G Lee, J. Seo, S. Lee, H. Jung, B. Cho, C. Lee, B. Kwak, J. Cha, D. Kim, J. Ann, H. Kim, K. Kim, " SiteQ:Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP" , In 10th Text REtrieval Conference, pp.437-446, 2001

[6] L.Hirschman, " Natural Language Question Answering:The View from Here" , Natural Language Engineering, 7(4), pp.275-300, 2001

[7] S. Harabagiu, M. Pasca, and S. Maiorano, " Experiments with open-domain with open-domain textual question answering." , In COLING-2000, pp-292.298, 2000

[8] A. Ittycheriah, M. Franz, W. Zhu, A. Ratnaparkhi, " IBM' s Statistical Question Answering System" , In 9th Text REtrieval Conference, pp.229-334, 2000

[9] Vlado Keselj, " Question Answering using Unification-based Grammar" , Advanced in Artificial Intelligence, AI 2001, volume LNAI 2056 of Lecture Notes in Computer Science, Ottawa, Canada, Springer, 2001

[10] M.A. Pasca, S.M. Harabagui, " High Performance Question / Answer" , In 24rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.366-374, 2001

[11] E.M. Voorhees, " The TREC question answering track" , Natural Language Engineering, 7(4), pp.361-378, 2001