백과사전 질의응답을 위한 구문정보기반 정답색인방법

김현진 이충희 오효정 왕지현 장명길
한국전자통신연구원 음성/언어정보연구부 지식마이닝연구팀
{jini, forever, ohj, jhwhang, mgjang}@etri.re.kr

A LF based Answer Indexing Method for Encyclopedia Question-Answering System

Hyeon-Jin Kim, Chung-Hee Lee, Hyo-Jung Oh, Ji-Hyun Wang, Myung-Gil Jang
Knowledge Mining Research Team, ETRI

요 약

본 논문은 정답 색인 방법을 이용하여 응답 속도가 빠르고 정확한 백과사전 질의응답 시스템을 구현하는 방법을 제안한다. 논문에서 제안한 정답 색인 방법은 대상 문서에서 160여 개의 정답 유형 범주에 해당하는 정답 후보를 인식하고, 정답 후보와 색인 범주에 속하는 키워드를 색인단위로 정의하여 저장하였다. 특히 용언정보에 대해서는 LF(Logical Form)단위로 색인하여 색인 정확도를 높였다. 정답 랭킹에서는 사용자 질문에서 각 단어별로 문장 성분, 단어 가중치 정보 등을 이용하여, 필수단어를 산정하고 이를 정답랭킹의 방법으로 활용하였다. 이러한 방법론은 용언 정보를 활용해야 효과적인 백과사전이라는 문서 도메인의 특성을 반영하고, 빠른 질문 응답 시간을 보장하는 백과사전 질의응답 시스템에 적합하다.

1. 서론

문서 정보 검색의 한계를 극복하기 위해서 질의응답기술이 많이 연구되어 지고 있다[1,2]. 본 논문에서는 2003년도에 개발한 AnyQuestion1.0(이하 AnyQuestion1.0) 이어 ETRI 지식마이닝연구팀에서 개발 중인 백과사전 대상의 질의응답 시스템인 AnyQuestion2.0(이하 AnyQuestion 2.0)(http://anyQ.etri.re.kr)을 소개하고자 한다. AnyQuestion 1.0은 백과사전 인물분야 질의응답 시스템으로 인물분야의 표제어에 관한 사용자 질문에 대해 질문의도를 파악하여 단답형의 정답을 제시하였다[3]. 또한 정답 추출 과정도 3단계로 IE 기법을 이용한 QA 기술과 단락기반 정보 검색을 이용한 QA기술을 접목하였다. 그러나, 3단계 정답 추출 기법에서도 단락기반 정보검색을 이용한 QA 부분은 실시간 QA기술에서 요구하는 빠른 질문 응답 시간을 보장하지 못하는 단점을 가지고 있었다. 즉, 질문이 입력되면 정보검색엔진을 통해서 단락을 추출하고, 단락 내에서 정답에 해당하는 부분을 다시 추정해야 하기 때문에 정답 추출 속도가 많이 소요되는 문제점이 발생하였다. 이런 문제점을 보완하기 위해, AnyQuestion2.0에서는 AnyQuestion1.0에서의 단락기반 정보검색을 이용한 QA 기술을 대신하여 정답 색인을 이용한 QA 기술을 접목하였다. 즉, 2.0에서는 개선된 IE기반 QA 기술과 정답 색인 기술을 이용한 QA기술이 사용되었다. 또한 2.0에서는 인물분야에서 전체분야로 컨텐츠 도메인이 확장되고, 정답 추출도 단답형 뿐 아니라 서술형 정답까지도 제시할 수 있는 기능이 추가되었다. 본 논문에서는 AnyQuestion2.0의 주요 기술 중에서도 정답 색인 방법을 이용한 QA 방법에 대해서 소개하고자 한다.

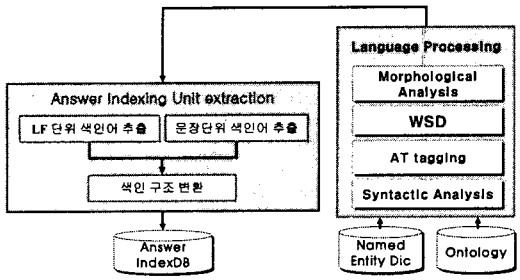
2. 관련 연구

국외(Trec 등)에서 연구되고 있는 QA을 보면, IE(Information Extraction)기법을 이용한 방법론[4]과 기존 정보검색 엔진을 응용한 단락 검색 시스템을 사용하는 방법론[5], 또 미리 정답 범주를 정의해서 문서를 태깅하여 색인(Predictive annotation)하는 방법론[6,7] 등이 있다. [4]는 IE 기법을 이용한 대표적인 QA 시스템으로 각 entity(예: person entity)들에 대해서 일종의 template(예: name, birth_time, what, when 등)을 정의하고, 정보추출을 통해서 각 template 값을 채우는 방식을 제안하고 있다. 이러한 IE 기법을 이용한 QA 시스템은 정답 제시에 속도가 빠르고, 신문 기사, 백과사전 등 특정 제한된 도메인에서는 효과적이라는 평가를 받고 있으나[5,8], entity 또는 template 정의에 있어서 일부 수작업의 노력이 필요하기 때문에, 확장성 면에서 문제가 제기되고 있다. 따라서 많은 국외 QA 시스템에서 따르는 방법론은 기존 정보검색 시스템에서 검색 단위를 문서가 아닌 단락 또는 문장 단위로 하여, 정답 후보가 포함된 일부를 검색 한 후에, 실시간으로 언어분석 등을 통해서 정답으로 추정되는 단어 또는 어구를 추출하는 단락 검색 응용 방법론을 채택하고 있다[5]. 그러나 이러한 방법론은 사용자 질문이 입력된 후 실시간으로 언어분석을 통해서 문장들을 분석해 내기 때문에 응답 시간이 매우 길다는 단점을 가지고 있다. 이런 단점을 극복하기 위해 [6,7]에서는 개체명 사전과 LSP(Lexico-Syntactic Pattern)을 이용하여 개체명을 인식하고, 이를 질의응답 시스템이 정답 가능한 후보로 미리 색인하는 방식을 이용하였다. [6]에서는 사용자의 질의유형을 105가지의 의미범주로 구분하고, 이에 따라 정답유형을 분류하였으며, Lexico-Syntactic Parser를 이용하여 사용자의 질의유형을 분석하여 색인된 정답DB에서 정답후보를 순위화하고 이를 정답으로 제시하였다. 하지만 [6]와 [7]에서는 정답후보를 중심으로 색인하는 키워드의 대상을 명사로 한정하고, 색인 범주도 한 문장 또는 인접 문장 등으로 고정하여, 그들사이의 구문적인 정보를 활용하지 않는다. 일반 웹 컨텐츠의 경우에는 이러한 고급정보를 활용하는 것이 어려울 수 있고 효과가 떨어질 수 있다.[6] 그러나 본 논문에서는 특정 도메인(백과사전 분야)을 대상으로 질의응답시스템을 구현하는데 있어, 구문 정보(Logical Form)가 포함된 색인 범주를 활용한 정답 색인 방법론이 좀 더 효과적이라는 것을 보여주고자 한다. 또한 정답추출에 있어서도, 기존 방법론들에서는 일반 정보검색의 질의처

리 방식을 응용한 것과 달리, 질문 내 구문정보 등 고급정보를 활용하여 정답 필터링을 시도한 방법론을 제시하고자 한다.

3. 정답 색인

정답 색인 과정은 입력된 문서에서 정답 후보를 인식하기 위해서 언어분석을 하는 부분과 인식된 정답 후보들과 2단계의 색인 범주 내의 키워드들을 추출하여 색인구조로 변경하여 저장하는 부분으로 크게 나뉜다. 정답 색인 과정을 간단히 도식화하면 [그림 1]과 같다.



[그림 1] 정답 색인 과정

논문에서 정의한 정답유형의 범주는 PERSON, STUDY_FIELD, THEORY, ARTIFACTS, ORGANIZATION, LOCATION, CIVILIZATION, DATE, TIME, QUANTITY, EVENT, ANIMAL, PLANT, MATERIAL, TERM으로 모두 15개의 대분류로 구성되어 있다. 이들은 각기 세부 범주를 가지게 되는데, 세부분류는 최하 3단계에서 최고 4단계로 세분화되어 있고, 총 정답유형 범주 수는 160여 개로 구성된다. 정답유형인식기에서 위에서 정의된 정답유형으로 각 문장에서 정답 후보를 인식하면, 정답색인구조 생성 파트에서는 추출된 정답 후보들과 동일한 색인 범주내에 있는 주변 키워드 들을 연결하여 정답색인구조로 생성하게 된다. 색인구조 추출은 다음의 두 단계에 걸쳐서 추출된다. 첫 단계는 LF(Logical Form) 단위로 정답색인구조를 생성하게 되는데, 여기서 LF는 용언 격틀을 이용하여 생성한 구문트리에서 각 용언과 구조적으로 연결된 노드들을 의미구조로 변경한 것으로 정의된다[3]. LF구조에서 연결된 노드들 중에 정답유형인식기에서 태깅된 노드는 정답후보집합(answer candidate words set)이 되고, 나머지 노드들 중에서 용언은 용언집합(verb set), 그 외 명사(복합명사, 속격어구 포함), 부사 등은 단서어집합(content words set)으로 분류된다. 이때 색인정보에는 LF의 노드정보(S:subject, O:object, V:verb, A:adverb)가 추가된다. 각 집합에 속한 단어들은 다음의 정답색인구조 생성 수식과 같이 생성되게 된다.

$$C\,(content\ words\ set) = \{c_1, c_2, c_3 \ldots\ldots\ c_l\}$$

$$V\,(verb\ set) = \{v_1, v_2, v_3 \ldots\ldots\ v_m\}$$

$$A\,(answer\ candidate\ words\ set) = \{a_1, a_2, a_3 \ldots\ldots\ a_n\}$$

$$AnswerIndexUnit = (c_i, a_j) \quad c_i \in C,\ a_j \in A$$

$$AnswerIndexUnit\_forVerb = (c_i, v_j, a_k) \quad c_i \in C,\ v_j \in V,\ a_k \in A$$

두 번째 단계는 문장 단위로 정답색인구조를 추출한다. 즉, 한 문장 내에서 단서어집합과 정답후보집합 사이에서 생성된 정답색인구조 중 앞 단계인 LF 색인 범주에 속하지 않는 것을 그 대상으로 한다. 색인정보로는 각 집합사이의 거리정보가 포함된다. 문장 단위 색인구조에서는 용언집합은 제외된다. 용언집합이 포함된 색인구조 생성은 LF단위로 한정 하였는데, 이는 용언의 경우에는 같은 문장내에 존재하더라도 구조적으로 연관성이 없는 경우에는 노이즈로 작용할 가능성이 많으므로 제한하여서 색인구조 과생성을 방지하였다. 예제 문장을 입력으로 처리하는 과정을 설명하면 다음과 같다.

[예문 1]
국제적십자사에서는 나이팅게일상을 마련하여 매년 세계 각국의 우수한 간호사를 선발, 표창하고 있다.(표제어: 나이팅게일)

[정답유형인식]
*<국제적십자사:OGG_SOCIETY>*에서는 *<나이팅게일상:CV_PRIZE>*을 마련하여 매년 세계 각국의 우수한 *<간호사:CV_OCCUPATION>*를 선발, 표창하고 있다. (*표제어: 나이팅게일:PS_NAME*)

[정답후보추출]
국제적십자사:*OGG_SOCIETY*
나이팅게일상:*CV_PRIZE*
간호사:*CV_OCCUPATION*

[LF 구조 추출]
마련하(<subj:국제적십자사<*OGG_SOCIETY*>가 > <obj:나이팅게일상<*CV_PRIZE*>를 >)
우수하(<subj:간호사 <*CV_OCCUPATION*>가 >)
선발하(<subj:국제적십자사< *OGG_SOCIETY* >가> <obj:간호사 <*CV_OCCUPATION*>를 >)

<LF 단위 정답색인구조 생성>

AnswerIndexUnit 생성	(나이팅게일상, 국제적십자사):Object-Subject (국제적십자사, 나이팅게일상):Subject-Object (국제적십자사,간호사):Object-Subject 등
AnswerIndexUnit_Verb 생성	(나이팅게일상, 마련하다, 국제적십자사):Verb-Subject (간호사, 선발하다, 국제적십자사):Verb-Subject 등

<문장 단위 정답색인구조 생성>

AnswerIndexUnit 생성	(세계+각국, 국제적십자사):Distance (간호사, 나이팅게일상):Distance (매년, 나이팅게일상) :Distance 등

4. 정답 제시

정답제시 방법은 사용자의 질문이 들어오면, 질문분석 모듈을 거쳐서 나온 결과를 입력으로 정답색인 모듈에서 저장한 색인 DB에서 해당 질문에 맞는 정답을 추출하고 제시하는 기능을 수행한다. 정답제시과정은 질문의 구조정보를 활용하여 질문 키워드 가중치를 부여하는 부분과 검색된 정답 후보를 순위화하는 부분으로 나뉜다.

4.1. 질문 키워드 가중치

논문에서는 질문 키워드 가중치를 구하기 위해, 먼저 질문을 구성하는 단어들 중에서 정답 추출에 가장 주요한 필수단어를 선정하도록 하였다. 필수단어 선정에 필요한 수식은 다음과 같다.

$$ScoreE(Se_i) = \sum_i w_i * sf_i$$

sf_1 = *Title point (백과사전 항목이름)*

sf_2 = *LF point* $\in$ *{subject point, object point, adverb point}*

sf_3 = *AT point* $\in$ *{PLO point, notPLO point}*

w_i = *each feature weight*

필수단어 선정에는 크게 타이틀정보 가중치(즉 질문의 키워드가 백과사전 항목 이름인 경우), LF구성요소 가중치(필수격 요소에 있는 경우 가중치 부여), AT가중치(정답유형 중에서도 PLO에 해당하는 경우는 그렇지 않은 정답유형에 비해서 높은 가중치 부여) 등 세가지 요소의 자질이 사용된다. 높은 점수를

가진 키워드가 필수 단어로 선정된다. 필수 단어 선정 후에는 각 키워드를 다음의 가중치 수식에 따라서 점수를 부여하게 된다.

if 필수단어

$$Q(Weight) = extra\_each\_weight + extra\_plus\_weight * (\frac{1}{max\_score\_count})$$

else

$$Q(Weight) = extra\_each\_weight - extra\_plus\_weight$$

max_score_count : 필수단어수
total_query_count : 질의에 나타난 키워드수
total_verb_query_count : 질의에 나타난 용언키워드수
*verb_weight = (total_verb_query_count / total_query_count) * Wv*
Wv : 용언 가중치 상수값
extra_total_weight : 1 - verb_weight
extra_each_weight = extra_total_weight / total_query_count
extra_plus_weight = max_score_count / total_query_count

질문 키워드 가중치가 정해지면, 질문에 나타난 키워드와 정답색인DB에 저장된 정답색인구조들 사이에 유사도를 계산하여 정답후보를 추출한다. 이때 유사도 계산은 p-Norm모델의 AND 오퍼레이션을 이용한다.

4.2. 정답 후보 순위화

정답랭킹에서 쓰이는 수식은 다음과 같다. 앞서 구한 질문 키워드 가중치를 이용한 1차 가중치 값과 다음의 요소들에 대해서 생기는 2차 가중치 값을 더한 것이 최종 가중치가 된다.

Document matching weight	질문에서 유추한 Document(백과사전 표제어 정보로 유추 가능함)에서 추출된 정답 후보인 경우 예) 박정희의 고향은 어디인가? (추천 표제어: 박정희 문서)
Distance weight	질의어 키워드와 정답후보간의 거리 가중치
Occurrence weight	일정 순위 내의 동일한 정답후보의 출현 빈도 가중치

$$Score(A_i) = \alpha * w_i + (1-\alpha) Score(R_i)$$

$$Score(R_i) = \sum_i sw_i * sf_i$$

$$w_i = \sum_i Q(w_i)$$

sf_1 = *Document matching weight*
sf_2 = *Distance weight*
sf_3 = *Occurrence weight*
sw_i = *feature weight*

5. 성능 평가 및 분석

AnyQuestion2.0을 객관적으로 평가하기 위해서, 평가 질의와 정답쌍으로 구성된 백과사전 질의응답 평가셋(총 402개)을 구축하였다[3]. 질의응답 시스템을 평가하기 위해 사용한 평가지수(measure)는 정답 역순위 평균(MRAR: Mean Reciprocal Answer Rank)이다. 정답 역순위 평균이란 사용자가 원하는 정답이 몇 번째에 나타났는가에 대한 순위를 평가하는 방법으로, 순위를 1/n의 가중치로 반영한다. 본 실험에서는 402개의 평가셋으로 상위 5등까지의 순위를 평가하였다. 실험은 단락검색을 활용한 AnyQuestion1.0의 성능(표2)과 본 논문에서 제안한 LF단위와 용언정보를 활용한 AnyQuestion2.0의 성능(표3)을 비교하였다. 그리고 이를 AnyQuesiton2.0에서 LF정보를 활용하지 않은 시스템(표4)과도 성능을 비교하였다.

[표2] Result of AnyQuestion1.0

	1	2	3	4	5
# correct answer	188	25	13	3	2
MRAR	0.51				

[표3] Result of AnyQuestion2.0 using LF

	1	2	3	4	5
# correct answer	185	25	10	3	1
MRAR	0.50				

[표4] Result of AnyQuestion2.0 not using LF

	1	2	3	4	5
# correct answer	149	38	15	4	5
MRAR	0.43				

[표2]과 [표3]를 비교하면 논문에서 제안한 색인방법론을 사용한 시스템이 AnyQuestion1.0과 비슷한 성능을 보이고 있다. 이것은 AnyQuestion1.0에서 질문응답 속도가 평균 5초(Pentium4 PC)를 넘은 데 반해, 제안한 방법론으로는 실제로 0.5초 이내 정답을 제시하는 점을 비교하면, 매우 유용하다는 것을 말해주고 있다. 또한 논문에서는 문장의 구조정보인 LF정보를 활용해서 정답을 추출하는 방법론을 제안했는데, [표3]와 [표4]을 비교하면, LF정보를 활용한 시스템의 성능(표3참조)이 확실히 높다는 것을 알 수 있다. 이는 백과사전 분야에서는 용언의 활용도가 높고 또한 문장 내의 구조정보를 활용하는 것이 정확도를 높여준다는 것을 보여주는 결과라고 볼 수 있다.

6. 결론 및 향후 연구방향

본 논문에서는 구문정보에 기반한 정답색인 방법론을 이용하여, 응답 속도가 빠르고 비교적 정확한 백과사전 질의응답 시스템을 제안하였다. 문장구조 정보는 도메인에 따라서는 오히려 성능을 저하할 수 있는 요소를 가지고 있으나, 논문에서 적용한 백과사전 분야에서는 오히려 구조정보에서 발생한 용언과 격정보가 정답 추출에 있어서 정확도를 높여주는 정보로 활용한다는 것을 알 수 있었다. 본 연구팀에서는 백과사전 전 분야에 대한 단답형, 서술형 정답을 제공하는 질의응답 시스템인 AnyQuestion2.0 개선과 함께 나열형 정답 처리를 포함하여 타 도메인으로 확장성을 고려한 실용적인 한국어 질의응답 시스템을 개발할 예정이다.

7. 참고 문헌

[1]Ellen M, Voorhees: Overview of TREC 2003 Question Answering Track. The Proceedings of the twelfth Text REtreival Conference(TREC-12), November 2003.

[2]Harabagiu S., Moldovan D., Pasca M., et al.: FALCON: Boosting Knowledge for Answer Engines TREC-9, 2000.

[3] H. J. Kim, H. J. Oh, C. H. Lee., et al.: The 3-step Answer Processing Method for Encyclopedia Question-Answering System: AnyQuestion 1.0. The Proceedings of Asia Information Retrieval Symposium (AIRS) (2004) 309-312

[4]Wei Li, Rohini K. Srihari : Extracting Extract Answers to Questions Based Structural Links, Coling-2002

[5]Sanda M. Harabagiu, Steven J. Maiorano: Finding Answers in Large Collections of Texts: Paragraph Indexing + Abductive Inference, AAAI-1999.

[6]Harksoo Kim, Jungyun Seo: A Reliable Indexing Method for a Practical QA System, Coling-2002.

[7]Prager J., Brown E. and Coden A.: Question-Answering by Predictive Annotation, The proceedings of SIGIR 2000.

[8]Julian Kupiec: MURAX: A Robust Linguistic Approach for Question Answering Using On-line Encyclopedia, SIGIR 93