

# 질의 응답 시스템에서 개체 피드백을 이용한 정답 추출

나승훈<sup>0</sup>, 강인수, 이상율, 이종혁  
 포항공과대학교 전자컴퓨터공학부 컴퓨터공학과, 첨단정보기술 연구센터  
 (nsh<sup>0</sup>, dbaisk, gilbert, jhlee)<sup>0</sup>@postech.ac.kr

## Answer Extraction Using Named Entity Feedback in Question Answering System

Seung-Hoon Na<sup>0</sup>, In-Su Kang, Sang-Yool Lee, Jong-Hyeok Lee  
 Div. of Electrical and Computer Engineering  
 Pohang University of Science and Technology

### 요약

질의 응답 시스템(Question Answering: QA)에서 정답 유형 분류(Answer Type Taxonomy: ATT)란 사용자 질문 분석을 위한 의미 분류 체계를 의미하는 것으로, ATT의 크기가 클수록 시스템의 성능은 높아진다. ATT를 확장하기 위해서는, 개체(Named Entity)에 의미 범주를 결정하는 개체 분류기(Named Entity Tagger)의 분류 체계가 세분되어야 하는데, 기존의 개체 분류기는 한 문서 내에서 그 개체의 분류를 시도하기 때문에, 분류를 위한 문맥 정보의 양이 부족하여, 정확하고 상세한 분류를 기대하기 힘들다. 본 논문에서는 동일 개체에 대한 문맥 정보를 수집하기 위해, 그 개체가 나타나는 다른 문서들을 검색하는 개체 피드백(Named Entity Feedback)이라는 기법을 사용한다. 개체가 상세히 분류됨에 따라 ATT도 확장될 수 있었으며, 이렇게 확장된 ATT 상에서의 정답 추출은 baseline보다 약 7%정도의 성능 향상을 보여, 개체 피드백의 효과를 확인할 수 있었다.

### 1. 서론

질의 응답 시스템(Question Answering System: QA)은 대용량의 데이터 모음으로부터 사용자의 다양한 자연어 질문을 입력으로 받아 문서가 아닌 정답을 제공해주는 시스템이다[13].

전형적인 QA시스템은 그림 1과 같이 3가지 컴포넌트로 나뉘어 기술될 수 있다[6].<sup>1</sup>

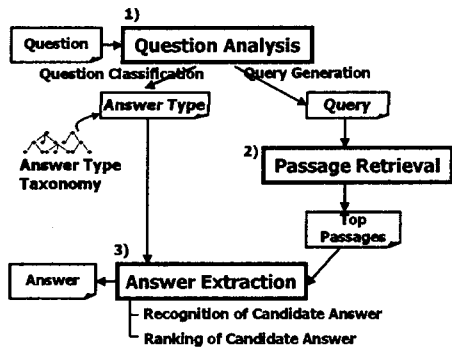


그림1: 전형적인 QA시스템의 구조.

1) 질문 분석(Question Analysis)단계: 질문 분류, 질의 생성  
 2) 단락 검색(Passage Retrieval)단계: 질문과 관련된 단락 집합 검색  
 3) 정답 추출(Answer Extraction)단계: 정답 후보 인식, 정답 후보 랭킹  
 위의 세가지 컴포넌트 중에서, 전체 QA시스템의 성능에 가장 큰 영향을 미치는 부분은, 3)정답 추출 단계이다[2][7]. 본 논문에서는, 정답 추출 단계의 성능을 향상시키기 위해, 개체 피드백(Named Entity Feedback)을 사용하여 ATT를 확장하는 방법을 제제안한다.

본 논문의 구조는 다음과 같다. 제 2장에서는, 정답 추출의 성능 향상을 위한 기존의 연구를 고찰하면서 개체 피드백 사용 배경에 대해 기술한다. 제 3장에서는 제안하는 방법에 대해 상세히 설명하고, 제 4장에서는 실험 결과를, 제 5장에서 결론을 언급할 것이다.

### 2. 관련 연구

정답 추출의 성능 향상을 위해서는, 정답 후보의 인식(Recognition of Candidate Answer)을 개선시켜 정답 후보의 개수를 줄이고, 또한 효과적인 정답 후보 랭킹 방법(Ranking Candidate Answer)을 사용하여 실제 정답이 상위로 올라가도록 해야 한다.

많은 QA시스템이 단순한 근접성(proximity)에 기반하여 정답 후보를 추출하고 있다[1][5][8]. 근접성 기반 정답 추출은, 질문의 단어들과 정답 후보간의 위치 차이를 최소화 시키도록 정답 후보를 랭킹하는 방법이다. 그러나, 이 방법은 자연어 분석을 하지 않아 빠르다는 강점을 가지고 있지만, 위치적인 정보만을 사용하기 때문에 그 한계를 벗어날 수 없다[10].

이러한 이유로, [2][7][11]에서는 구문 분석과 같은 보다 정교한 자연어 처리를 사용하여 근접성 기반의 정답 랭킹의 한계를 극복하고자 하였다. 이 방법은 정답 후보와 질문내의 단어간의 구문 관계 제약을 부여하여, 그 제약이 일치하는 정보만을 정답 후보로 인식하거나, 가산점을 부여한다. 그러나, 구문 분석에 기반한 정답 추출 방법은, 높은 성능을 갖는 구문 분석기의 부재와, 같은 의미에 대해 서로 다른 구문적 말바꿈(paraphrasing) 현상으로 인해, 성능 향상을 기대하기 힘든 상황이다.

<sup>1</sup> 여기서 기술한 전형적인 시스템의 구조는 필자가 지금까지 여러 다른 QA시스템을 분석하여 일반화한 것이며, 시스템의 세부 사항은 조금씩 차이가 있을 수 있음을 밝혀 둔다.

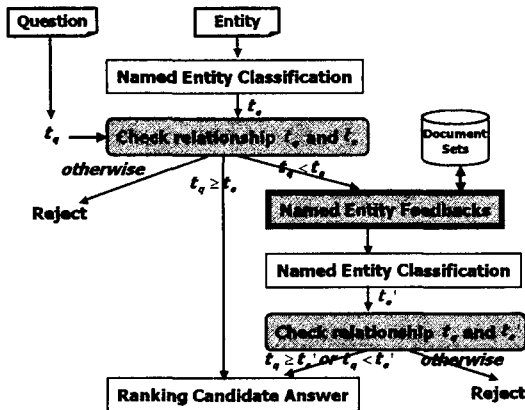


그림 2: 개체 피드백을 이용한 정답 후보 인식

한편, 정교한 정답 유형 분류(Answer Type Taxonomy: ATT)를 구축하여 정답 후보의 개수를 줄이는 방법은 높은 성능을 보이고 있다 [3][12]. 왜냐하면, 시스템이 다양한 ATT를 가지게 된다면, 질문을 더 많이 처리할 수 있게 되어, 정답 후보 인식의 적용률이 높아지게 될 것이고, 정교한 ATT를 가지게 된다면, 정답 후보의 인식의 정확률이 개선되어 전체적으로 시스템의 성능이 향상 되기 때문이다. 본 논문 역시, ATT의 확장으로 정답 추출의 성능을 개선시키고자 한다.

그런데, 정교한 ATT를 구축하기 위해서는, 개체(Named Entity)에 의미 범주를 결정하는 개체 분류기(Named Entity Tagger)의 분류 체계가 세분되어야 한다. 그러나 기존의 개체 분류 방법에서는 만약 그 개체가 개체 인스턴스 DB에 없는 경우, 그 개체가 출현한 특정 문서의 하나의 국소 문맥(local context)만을 개체를 분류할 수 있는 유일한 정보로 사용한다. 따라서 그 개체의 분류 한계는 그 국소 문맥이 주는 정보에서 크게 벗어 날 수 없다. 그러나 하나의 개체는 특정 문서에만 있는 것이 아니라, 시스템이 대상으로 하는 문서 집합 전체에서 여러 번 나타나므로, 실제로 그 개체에 대한 국소 문맥의 정보는 굉장히 많다. 기존의 QA시스템에서는, 이러한 국소 문맥 정보의 분산성을 간과하여, 이미 개발된 개체 분류기를 단순히 덧붙여 사용하고 있기 때문에, 개체 분류 체계가 더 세분화되기 어렵다.

따라서 우리는 이러한 사실에 입각하여, 문서 내의 개체가 정답 후보인지를 판별할 때, 그 개체가 나타나는 다량의 문서들로부터 국소 문맥을 수집하여, 개체 분류의 성능을 향상시키고자 한다. 이를 위해서는 개체가 다시 검색 엔진의 질의로 들어가야 하기 때문에, 이 과정을 우리는 개체 피드백(Named Entity Feedback)이라고 명명하였다.

개체 피드백을 사용하여, 더 세밀한 개체 분류가 가능해짐에 따라 우리는 ATT를 WordNet과 같은 범용 시소러스까지 확장시킬 수 있게 되었다.

3. 개체 피드백을 이용한 정답 후보 인식

개체 피드백을 기술하기 위해, 본 논문에서 사용되는 기호와 식은 아래와 같다.

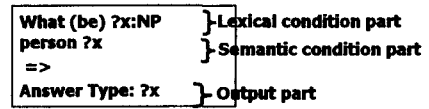


그림 3: 어휘 의미 패턴의 예

$\mathcal{T}$ : 정답 부류 체계(Answer Type Taxonomy)

$t_q \in \mathcal{T}$ : 질문이 요구하는 정답 유형(Answer Type)

$t_e \in \mathcal{T}$ : 개체(Named Entity)의 부류

$t_e' \in \mathcal{T}$ : 개체 피드백을 통해 이루어진 개체에 대한 상세 부류

$t_q < t_e$ :  $t_q$ 가  $t_e$ 의 하위어 또는  $t_e$ 가  $t_q$ 의 상위어일 때 참

개체 피드백을 사용한 정답 후보 인식 절차는 그림 2와 같다. 우리가 정답 후보임을 판단하고자 하는 각각의 개체에 대해, 우선 개체가 처음 출현했던 단락의 국소 문맥만을 가지고, 그 개체의 부류  $t_e$ 를 구하고, 그것이 질문의 정답 유형(Answer Type)에 속하는지를 테스트한다. 현재까지의 국소 문맥만 가지고 그 개체를 정답 후보인지를 판별하는데 충분하다면, 개체 피드백은 이루어지지 않으며 그 다음 절차를 진행하게 된다. 그러나 국소 문맥이 불충분하다면( $t_q < t_e$ 인 경우), 개체 피드백(Named Entity Feedbacks)을 통해 다량의 국소 문맥을 획득한 후 그 개체에 대한 상세 부류  $t_e'$ 를 구하게 되고, 이  $t_e'$ 를 사용하여 정답 후보에 대한 인식을 다시 시도한다.

다음 각 절마다 각각의 절차에 대해 상세히 설명하도록 하겠다.

3.1 정답 유형의 결정 (Determine Answer Type)

우리는 ATT로 WordNet을 사용한다.<sup>2</sup> 정답 유형은 질문내의 특정 단어(focus word)로부터 결정될 수 있는데, 이것을 찾아내기 위해, 수동으로 구축된 약 20여개의 어휘-의미 패턴(Lexico-Semantic Pattern)을 사용하였다. 어휘-의미 패턴의 구조의 예는 그림 3에 나와 있으며, 이 규칙에 질문 "What is the governor of Colorado?"를 적용시키면, governor<sup>3</sup>를 정답 유형으로 추출하게 된다.

3.2. 개체 분류(Named Entity Classification)

개체 분류를 위해, 우리는 person, location, organization으로 분류되어 있는 약 10만개 정도의 개체 인스턴스를 보유하고 있다. 일단 개체가 인스턴스 DB에 속하게 되면, 개체를 위의 3가지 중 하나로 분류할 수 있게 되지만, 더 세부적인 분류를 위해서는 국소 문맥을 참조해야 한다.

우리는 간단한 언어 규칙을 사용하여, 그림 4와 같이 해당 개체의 다양한 국소 문맥 정보를 인식한다[9]. 국소 문맥 중 특히 동격이나 자격과 같은, 개체의 유형을 결정하는 특정 단어가 나타나면, 개체를 그 단어가 속하는 시소러스 개념으로 상세히 분류할 수 있다. 그림 4의 예에서 Kenneth Starr의 경우는 person보다 상세히

<sup>2</sup> 물론, 문서에 나타나는 개체를, WordNet상의 한 개념(최소개념)으로 정확히 분류할 수 있는 개체 분류기를 만들었다는 뜻은 아니다. 우리는 목적상, 되도록이면 상세한 분류 체계를 가지는 것을 원했으며, WordNet은 그러한 우리의 목적에 매우 적합했다.

<sup>3</sup> 여기서 governor는 WordNet의 governor에 대한 첫 의미의 synset이다.

Context Info	Example
Appositive	1) <b>Nancy Powers</b> , a sales director for a medical company 2) his superstar wife <b>Nicole kidman</b> 3) <b>Steve Thomas</b> , the police detective
Role	1) <b>Mrs. Aquino</b>
Relative Clause	1) <b>Kenneth Starr</b> , who
Preposition	1) in <b>Washington</b>

그림4: 개체 분류를 위해 사용하는 국소 문맥 정보의 예

시소러스 관계	1) $t_q \geq t_c$	2) $t_q < t_c$	3) $t_q < t_c$
정답 후보 여부	확실한 정답 후보	정답 후보 대상에서 제외	정답 후보인지 불확실: 더 많은 국소 문맥 필요

그림5: 개체 부류와 정답 유형간의 시소러스 위상에 따른 정답 후보 여부 결정  
분류될 수 없지만, Steve Thomas의 경우는 동격 정보를 활용하면, detective로 분류될 수 있다.

### 3.3. 정답 후보 여부 판별

정답 후보 여부는 개체 부류와 정답 유형간의 시소러스 관계에 따라 이루어지며, 그림 5에 간략히 기술되어 있다. 시소러스 관계가 3)에 해당할 때, 우리는 개체 피드백을 사용하여 시소러스 관계가 1)이나 2)가 되도록 할 것이다. 개체 피드백을 사용해도 여전히 3)의 관계를 갖는다면, 비교적 낮은 점수를 주고 정답 후보로 간주하게 된다.

### 4. 실험

우리는 TREC8-TREC10 QA track의 질문 중에서, 문서 상에 정답을 포함하면서 개체를 묻는 질문 300개를 가지고, 개체 피드백을 사용한 QA시스템의 성능을 평가해 보았다. 우리는 개체 피드백으로 검색된 문장 중 임의의 100개 문장으로부터 국소 문맥을 추출하였다. 그림 6은 개체 피드백을 사용한 정답 추출과, 개체 피드백이 없는 baseline과의 정답 추출의 성능을 비교한 것이다. 그림 6의 결과에서, PR은 상위 20개 단락 안에 정답이 포함되는 확률이며, Base-AE와 EF-AE는 각각 개체 피드백을 사용하지 않는 정답 추출과 개체 피드백을 사용한 정답 추출의 MRR이다.

실험 결과에서 알 수 있듯이 개체 피드백을 사용한 정답 추출의 성능이 기본 baseline보다 약 7%의 성능 향상을 보여, 제안한 방법이 효율적임을 알 수 있다.

### 5. 결론

본 논문에서는 질의 응답 시스템의 성능 향상을 위해, 개체 피드백을 사용한 개체의 국소 문맥 획득 전략을 제안하였고, 실험 결과로부터 우리의 접근 방법이 타당하다는 것을 알 수 있었다.

우리의 개체 분류 방법은 소규모의 개체 인스턴스 DB와 간단한

Answer Type	Percentage	PR	Base-AE	EF-AE
Person	48.98%	91.66%	0.476	0.483
Location	40.82%	85.00%	0.287	0.341
Organization	10.20%	80.00%	0.500	0.500
Total	100.00%	85.71%	0.401	0.427

그림6: 실험 결과. PR은 상위 20개 단락 안에 정답이 포함되는 확률이며, BaseAE와 EF-AE는 각각 Baseline정답 추출과 개체 피드백을 사용한 정답 추출 방법의 MRR이다. 언어 규칙만으로 이루어진 매우 간단한 방법이다. 개체 피드백을 이렇게 경량급의 개체 분류 방법만으로도 효율적인 정답 추출이 이루어질 수 있음을 보였다는 점에서 큰 의의가 있다고 하겠다.

감사의 글

본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았다.

### 참고 문헌

- [1] S.Abney, M.Collins, A. Singhal, "Answer Extraction", In Sixth Applied Natural Language Processing Conference, 2000
- [2] S.Buchholz, W. Daelemans, "Complex Answers: a case study using a WWW question answering system", Natural Language Engineering, 7(4), pp.301-323, 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] A. Miheev, C. Grover, M.Moens, "Description of the LTG System Used for MUC-7", Proceedings of the MUC-7, 1998
- [10] S.Na, I.Kang, O.Kwan, J.Lee, "Answer Candidate Ranking based on Syntactic Proximity in Question Answering", Proceedings of the 29th KISS Sprint Conference, pp 478-480, 2002
- [11] 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
- [12] 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
- [13] E.M. Voorhees, "The TREC question answering track", Natural Language Engineering, 7(4), pp.361-378, 2001