Semantic parsing 기반 지식 베이스 질의응답 시스템의 어휘-의미 패턴 질의 템플릿을 통한 보완

심효섭○, 박선영, 이근배
포항공과대학교 컴퓨터공학과
{hyosupshim, sypark322, gblee}@postech.ac.kr

Assisting semantic parsing-based QA system with lexico-semantic pattern query template

Hyosup Shim○, Seonyeong Park, Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Science and Technology

요 약

본 논문에서는 semantic parsing과 사전 정의된 어휘-의미 패턴 질의 템플릿 방법론을 결합하여 자연어 질의로부터 RDF 지식베이스에 질의하기 위한 SPARQL 쿼리를 생성하는 방법을 제안한다. semantic parsing 접근법은 문장의 표현과 분리된 형식적 의미표현만을 포착해내므로, paraphrase 혹은 의미 변화와 무관한 어순의 변화에 강인하지만, 일부 자연어 질의문장에는 단순한 의미 및 구조를 갖는 문장도 적합한 형식적 의미표현을 생성하지 못하는 단점이 있다. 따라서 이 연구에서는 이러한 단순한 문장에 있어서는 사전 정의된 질의 템플릿을 사용하여 적합한 쿼리를 생성하되, 적합한 템플릿을 선택하는데 있어 해당 질의문장의 어휘-의미적 유형을 포착하고 해당 정보를 이용하는 방법을 이용하였으며 이를 통해 주 방법론의 약점을 보완하는 제한적인 효과를 얻을 수 있었다.

주제어: semantic parsing, 지식베이스 기반 질의응답, 어휘-의미 패턴

1. 서론

최근 Freebase, DBpedia[1] 등 크라우드 소싱을 통해 구축된 대규모 RDF 기반 지식베이스(이하 KB)가 잇달아 일반에 공개되면서 이러한 KB를 활용하기 위한 방안이 다양하게 연구되고 있다. 특히 그 중에서도 KB에 질의하기 위해 요구되는 표준에 대한 이해 등 기술적 장벽을 해소하고 일반 유저의 KB 활용을 활성화하기 위한 방안으로서 자연어 인터페이스에 대한 연구가 주목을 받고 있다.

일반적으로 KB 자연어 인터페이스 시스템이 취하는 전략은 다음과 같다. 1) 질의 분석. 2) 질의 문장 내 엔티티와 술어를 KB 내 개념어휘로 매핑. 3) 매핑 결과를 통해 KB에 질의할 질의를 생성한 뒤 4) 생성된 쿼리로 KB에 질의하여 얻은 결과를 제공[2].

이 중 질의 문장 내 엔티티와 술어의 매핑이 시스템의 성능에 중요도가 높으며 관련 연구에서도 다양한 접근법을 취하고 있다.

본 논문에서는 semantic parsing을 중심으로 하고 어휘-의미 패턴 규칙에 기반한 query template을 통해 보완하는 KB 자연어 질의 시스템을 제안하고자 한다. 2장에서는 배경 연구 및 관련 연구를 간략하게 소개하고, 3장에서는 연구에 활용한 semantic parser에 대한 소개와 어휘-의미 패턴 규칙 및 그를 활용한 질의 처리과정을 설명한다. 4장에서는 설명된 방법으로 보완한 자연어 인터페이스 시스템의 성능을 평가하고, 5장에서 결론을 맺는다.

2. 관련 연구

2.1. semantic parsing

상술한 바와 같이 KB 자연어 인터페이스 시스템은 질의 분석 뒤 엔티티와 술어를 KB 내 개념 어휘에 매핑하는 전략을 일반적으로 취한다. semantic parsing은 본래 자연어 문장을 형식적이고 논리적이며 명시적인 의미로 t 실행될 수 있는 형식적 의미 표현으로 변환하는 과업으로,[3]. semantic parsing의 결과인 형식적 의미 표현에는 특정한 KB의 개념 어휘가 사용되게 된다. 초기 연구는 장난감 수준의 소규모 KB를 채용하였으나, 최근 Freebase 등의 대규모 KB를 기반으로 한 semantic parsing 연구 결과가 발표되면서[4], 특정 KB를 기반으로 하는 semantic parser를 해당 KB에 대한 질의 분석

및 KB 개념 어휘 매핑 과정으로 차용하는 연구가 발표되었다. [5]

2.2. KB 자연어 인터페이스 관련 연구

이전 연구에서는 엔티티와 술어의 매핑에 있어서 어휘 의미망과 문자열 유사도에 의존한 접근법을 주로 채용하였다. Querix는 WordNet synset을 이용하였고[6], PANTO는 문자열 유사도와 WordNet 의미망을 활용하였다[7]. QACID는 자연어 질의와 SPARQL 쿼리의 쌍으로 이루어진 데이터를 통해 bag-of-word 기반 지도학습 및 문자열 편집거리 기반의 술어 매핑을 수행하였다[8].

Unger 외 는 질의 문장을 파싱하여, 사전에 만들어 둔 도메인 독립적 어휘 사전과의 일치 정보 및 part-of-speech 패턴 정보를 기반으로 query template을 선택한 뒤, 도메인 의존적 어휘를 중심으로 template의 슬롯 정보를 채워 쿼리를 생성하는 전략을 취하였다[9].

대규모 KB를 기반으로 한 semantic parser가 사용가능해지면서 semantic parsing을 기반으로 한 접근법도 활용되고 있다. Berant은 대량의 코퍼스로부터 자연어구 - KB 술어 매핑을 만든 뒤 질의 문장에 beam parsing을 수행하면서 가장 적합한 부분 문장을 역시 가장 적합한 KB 개념 어휘로 교체하는 방법으로 형식적 의미표현의 후보를 생성한 뒤 이를 평가하는데 사전 구축한 paraphrase 모형을 활용하였다[10].

3. 방법

3.1. semantic parser

본 논문에서 사용한 semantic parser는 [Berant 외, 2013]의 방법을 채용하였으며, parser의 처리 결과인 형식적 의미표현을 SPARQL 쿼리로 변환하는 부분만을 별도로 구현하였다.

3.2. 어휘-의미 패턴 질의 템플릿

어휘-의미 패턴 질의 템플릿은 크게 문장 패턴 규칙, 슬롯 정보 템플릿, 질의 템플릿의 세 부분으로 나뉜다.

표 1 어휘-의미 패턴 질의 템플릿 예시

```
Sentence What VP#_VBZ NP#^NNP NP#_NNP
Slot:?x,selected_var,First
Slot:?p,class,Second
Slot:?y,resource,Third
Template:?y,?p,?x
```

문장 패턴 규칙은 어휘 패턴, chunk 유형 패턴, chunk 내 패턴으로 구성되어 있다. 어휘 패턴은 직접적인 어휘의 존재 여부를 통해 일치를 판단하며, chunk 패턴은 해당 문장을 청킹한 결과로 얻은 chunk의 수 및 유형을 통해 일치를 판단한다. chunk 내 패턴은 chunk 내의 요소 중 패턴에 포함된 part-of-speech에 해당하는 어휘가 존재 여부를 통해 판단한다. 표 1의 예시에서 첫 번째 줄이 문장 패턴 규칙에 해당한다. 어휘 패턴은 문장 첫 단어로 'What' 이 출현할 것을 요구하며, chunk 유형 패턴은 문장의 나머지 부분이 세 개의 chunk로 이루어져 있고, 각각의 chunk는 VP, NP, NP 유형을 가질 것을 요구한다.

슬롯 정보 템플릿은 질의 템플릿에 출현한 variable의 유형 정보와 매핑을 위한 정보를 추출하는 chunk의 식별자로 구성되어 있다. 예시에 언급된 두 번째 슬롯 정보 템플릿 "Slot:?x,class,Second"를 예로 들면, 슬롯 x는 엔티티의 유형 정보에 속하며, 이 슬롯을 매핑하기 위한 정보는 질의 문장의 두 번째 chunk에서 추출해야 함을 가리킨다.

마지막으로 질의 템플릿은 완성된 SPARQL 쿼리가 수행될 때 매칭을 수행하게 될 그래프 패턴을 나타낸다. 예시의 질의 템플릿은 완성된 SPARQL 쿼리가 세 개의 슬롯 "?y ?p ?x" 로 구성된 단일 트리플의 일치를 시도하는 쿼리가 될 것임을 보이고 있다.

3.3. 자연어 질의 분석 및 템플릿 선택

자연어 질의 문장은 질의 분석 과정을 통해 어휘-의미 패턴 질의 템플릿에 포함된 문장 패턴 규칙과의 일치를 확인하고 적합한 템플릿을 선택하기 위한 과정을 거친다. 이러한 질의 분석 과정에는 ClearNLP[11]의 POS tagging 도구와 OpenNLP[12]의 chunker 도구를 활용하였다.

표 2 문장 패턴 규칙과의 매칭 방법

```
if not matchLexicalPattern():
  return false
if not agreeInNumberOfChunks():
  return false
set numChunks <- number of chunks
for i = 1 to numChunks:
  if chunkTypeInSentence[i] !=
       chunkTypeInPattern[i]:
     return false
  for word in chunkInSentence[i]:
     for forbiddenPos
        in chunkInPattern.forbiddenPos():
       if pos(word) == forbiddenPos:
         return false
     for requiredPos
        in chunkInPattern.requiredPos():
       if pos(word) != requiredPos:
         return false
return true
```

3.4. 템플릿 적용

자연어 질의와 일치하는 문장 패턴 규칙을 가진 어휘-의미 패턴 템플릿을 찾았다면 일치하는 템플릿의 슬롯 정보 템플릿에 자연어 질의의 chunking 분석 결과로부터 해당 chunk가 포함하는 문자열을 포함하여 쿼리 템플릿으로부터 실제 SPARQL 쿼리 생성시 KB 개념어휘를 매핑하기 위한 템플릿 적용을 수행한다.

3.5. SPARQL 쿼리 생성

필요한 정보가 갖추어진 템플릿으로부터 KB 개념어휘에 슬롯정보를 매핑하고 그로부터 실제 수행이 가능한 SPARQL 쿼리를 생성한다.

SPARQL 쿼리 생성은 다음과 같은 과정을 거친다.

표 3 템플릿으로부터 KB 어휘 매핑 및 SPARQL 생성 과정

1. class, resource 타입을 가진 slot을 chunk 문자열의 유사도에 따라 실제 resource / class URI 에 매핑.
2. 매핑이 완료된 resource / class URI가 포함된 triple로부터 predicate URI의 후보를 인출.
3. property 타입을 가진 slot의 chunk 문자열을 기준으로 predicate URI 후보의 의미적 유사도[13]를 계산하여 랭킹을 매긴다.
4. 의미적 유사도가 가장 높은 술어 URI를 선택하여 매핑을 마친 뒤 실제 SPARQL 질의를 생성한다.

4. 실험 및 결과

4.1. 대상 질의문장

CLEF QALD 챌린지 다중 언어 QA 트랙의 영문 질의문장을 수집하여 인물과 관련된 질의를 추려내었다. 이를 다시 정답 SPARQL 쿼리의 트리플 수가 2개 이하인 질의로 제한하여 질의문장 146개를 테스트 셋으로 사용하였으며, 다시 이 중 정답 쿼리가 단일 트리플을 갖는 문장에 한하여 어휘-의미 패턴 질의 템플릿을 작성한 뒤, semantic parsing 방법론을 단독으로 적용하였을 경우와 어휘-의미 패턴 질의 템플릿을 함께 적용한 경우로 나누어 accuracy 측정을 시행하였다.

4.2. 결과

표 4. 실험 결과

	# of correct	accuracy
w/o LSP	70	0.4794
w/ LSP	79	0.5411

문장에서 엔티티 혹은 술어로 판단되는 부분의 분할을 전적으로 chunking 도구에 의존하고 있기 때문에 이러한 분할이 정교하지 못할 경우 분할된 문자열의 유사도에 의해 수행되는 엔티티 매핑의 성능이 제한되며, 이러한 제한점이 실험 결과로 나타난 것으로 생각된다.

5. 결론

어휘-의미 패턴 질의 템플릿은 주 방법론이 간단한 문형이나 의미에 대해서 잘 대응하지 못하는 경우에 주 방법론을 보완하기 위한 방안으로 고안되었다. 수동으로 작성된 템플릿 및 매칭 규칙을 필요로 하기 때문에 대규모로 적용하기에는 제한이 따른다. 하지만 사용자 질의의 빈도가 높고 질의를 처리하는 주 방법론이 대응하기 어려운 경우의 보완책으로서는 제한적이지만 나름의 역할을 수행할 수 있을 것으로 판단된다.

Acknowledgement

본 연구는 한국연구재단[NRF-2014R1A2A1A01003041, 다중화자 예측기반 지식강화 자연어 대화 시스템 기술 개발]과 우수기술연구센터 사업[10048448, 링크드 데이터 기반 대화형 질의응답 검색 프레임워크 개발]의 일환으로 수행하였음

참고문헌

[1] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, Christian Bizer: DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. To appear in the Semantic Web Journal, 2014.

[2] 박선영, 심효섭, 이근배, ISOFT at QALD-4: Semantic similarity-based question answering system over linked data, Unpublished Article, 2014.

[3] Zelle, John M., and Raymond J. Mooney. "Learning to parse database queries using inductive logic programming." Proceedings of the National Conference on Artificial Intelligence. 1996.

[4] Berant, Jonathan, et al. "Semantic Parsing on Freebase from Question-Answer Pairs." EMNLP. 2013.

[5] Yih, Wen-tau, Xiaodong He, and Christopher Meek. "Semantic Parsing for Single-Relation Question Answering.“Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014.

[6] Kaufmann, Esther, Abraham Bernstein, and Renato Zumstein. "Querix: A natural language interface to query ontologies based on clarification dialogs."5th International Semantic Web Conference (ISWC 2006). 2006.

[7] Wang, Chong, et al. "Panto: A portable natural language interface to ontologies." The Semantic Web: Research and Applications. Springer Berlin Heidelberg, 473-487. 2007.

[8] Ferrández, O., Izquierdo, R., Ferrández, S., & Vicedo, J. L. Addressing ontologybased question answering with collections of user queries. Information Processing & Management, 45(2), 175-188, 2009.

[9] Unger, Christina, et al. "Template-based question answering over RDF data." Proceedings of the 21st international conference on World Wide Web. ACM, 2012.

[10] Berant, Jonathan, and Percy Liang. "Semantic parsing via paraphrasing." Proceedings of ACL. 2014.

[11] http://clearnlp.wikispaces.com/

[12] http://opennlp.apache.org/

[13] Gabrilovich, Evgeniy, and Shaul Markovitch. "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis." IJCAI. Vol. 7. 2007.