



OKBQA-4

TUTORIAL : 16-17
SYMPORIUM : 18
HACKATHON : 18-21
July 2016, Jeju, Korea



오픈 지식베이스와 질의응답 협력개발 플랫폼: OKBQA-4

최기선 (KAIST)

OKBQA-4 Team:

김진동(DBCLS), Christina Unger(Bielefeld), Axel Ngonga (Leipzig), André Freitas (Passau), 함영균, 김지성, 남상하,
김정욱, 최규현 (KAIST)

2016. 11. 25



UNIVERSITÄT LEIPZIG



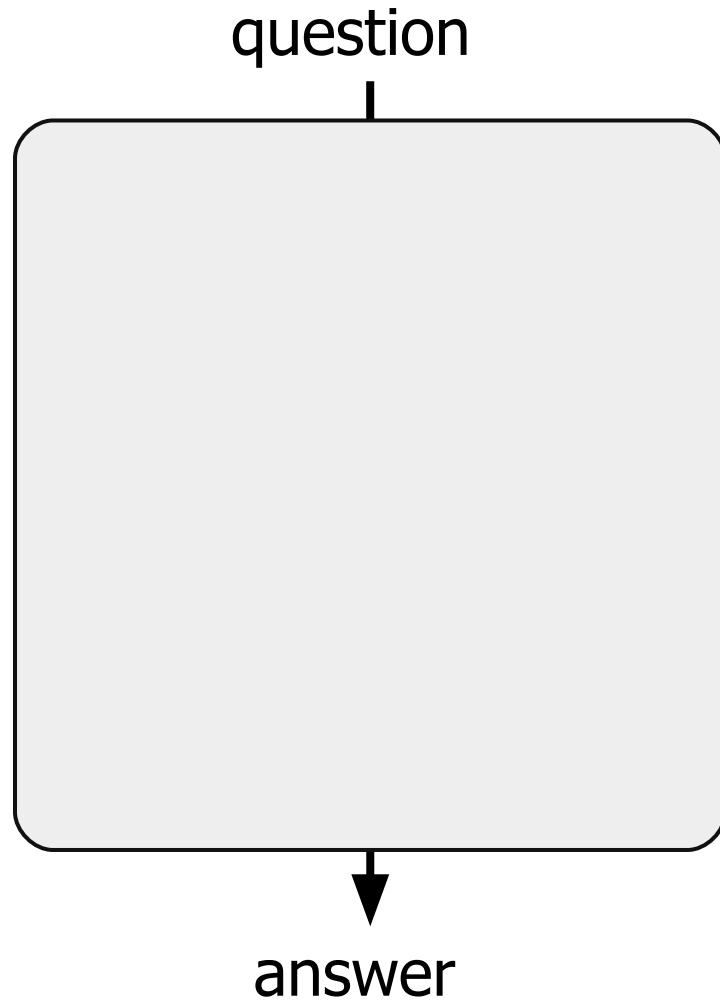
Agenda

1. Introduction
2. Components
3. Principles
4. Resources
5. Conclusions

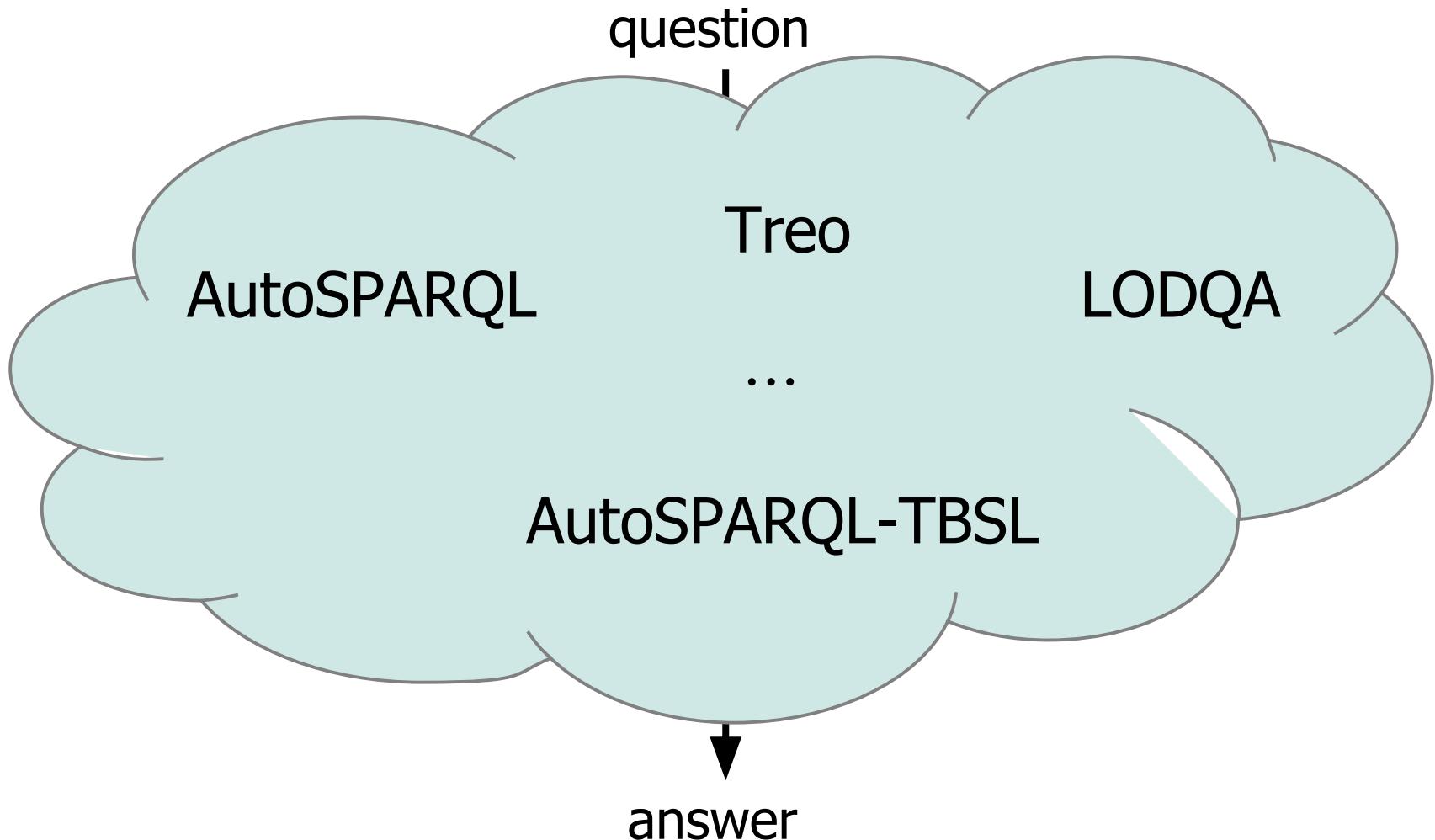


Introduction

Question-Answering

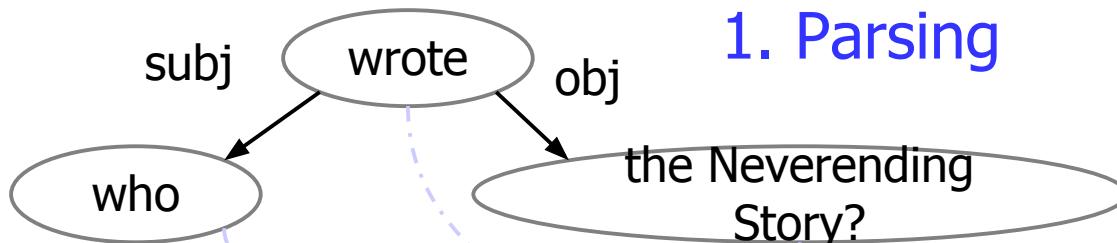


Existing Systems



Typical Problems

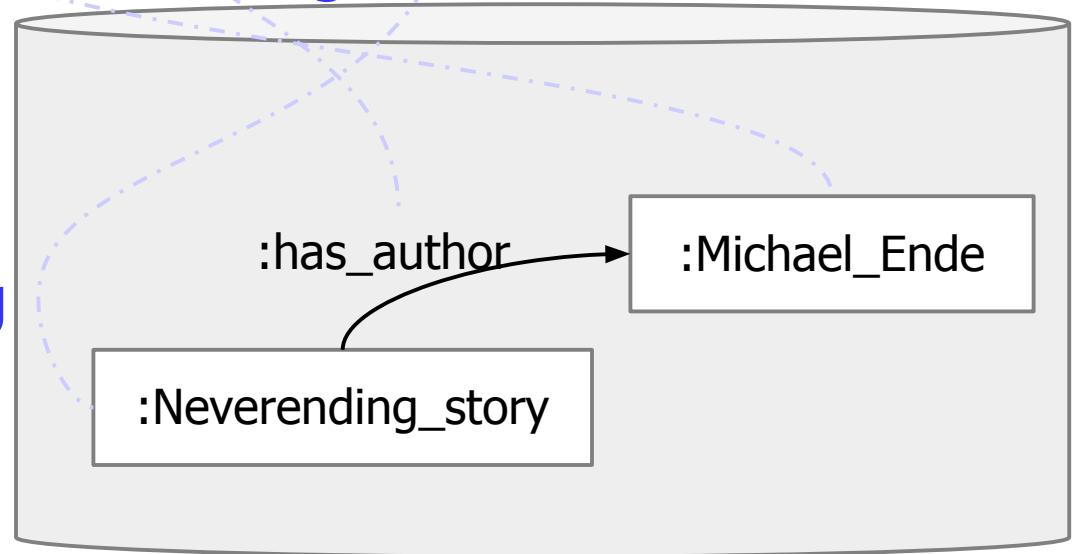
Who wrote the Neverending Story?



1. Parsing

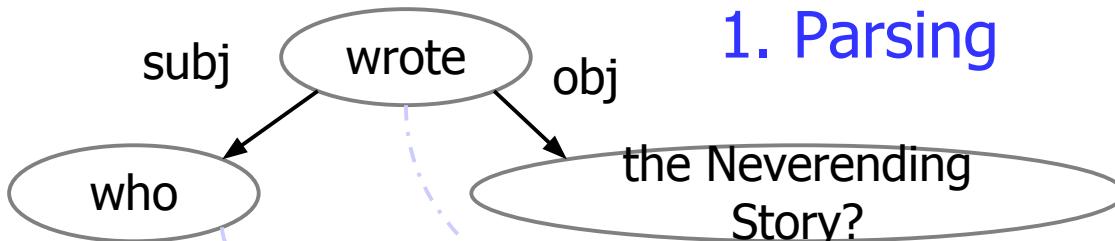
2. Lexical matching

3. Structural matching



Typical Problems

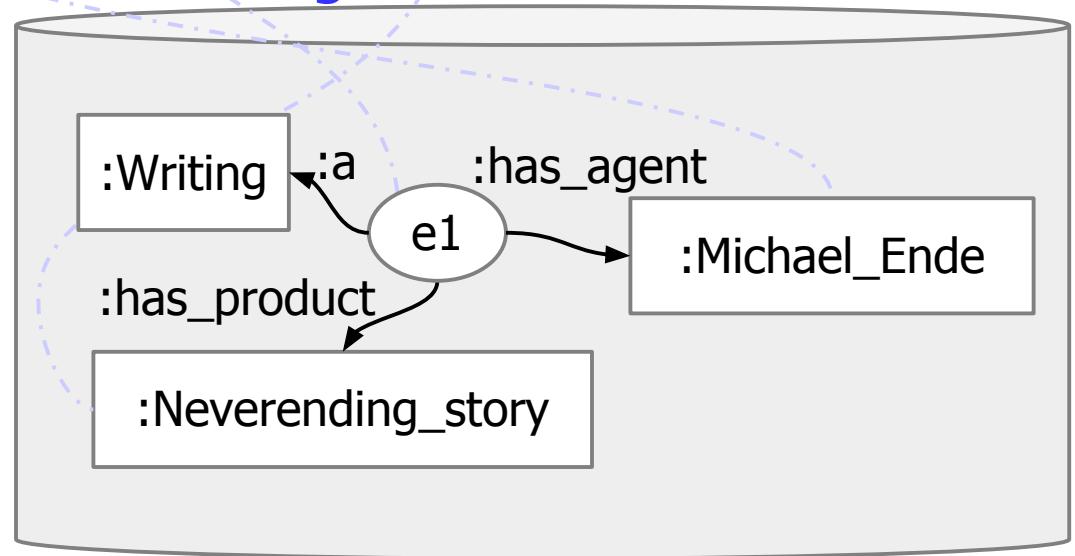
Who wrote the Neverending Story?



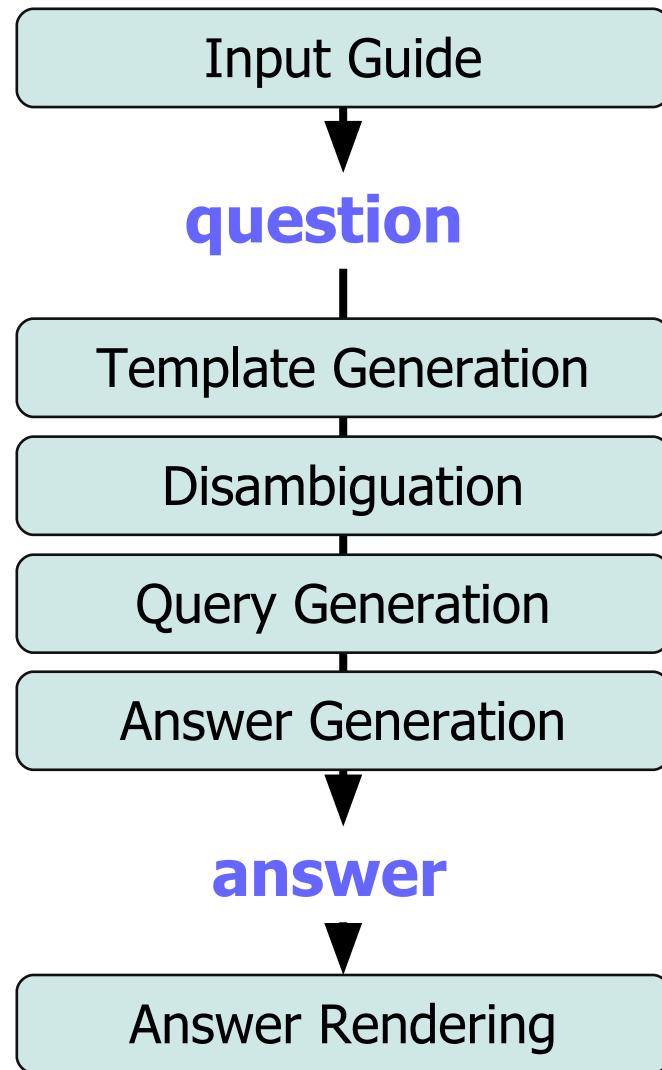
1. Parsing

2. Lexical matching

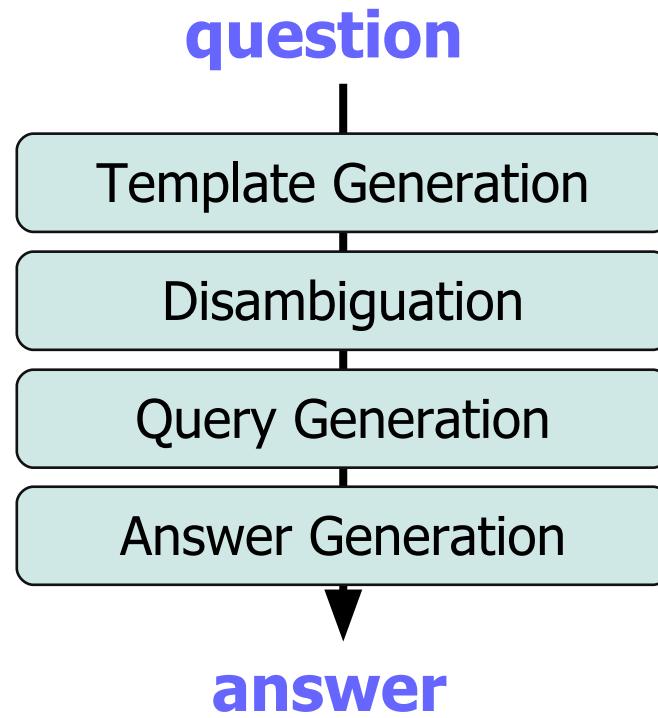
3. Structural matching



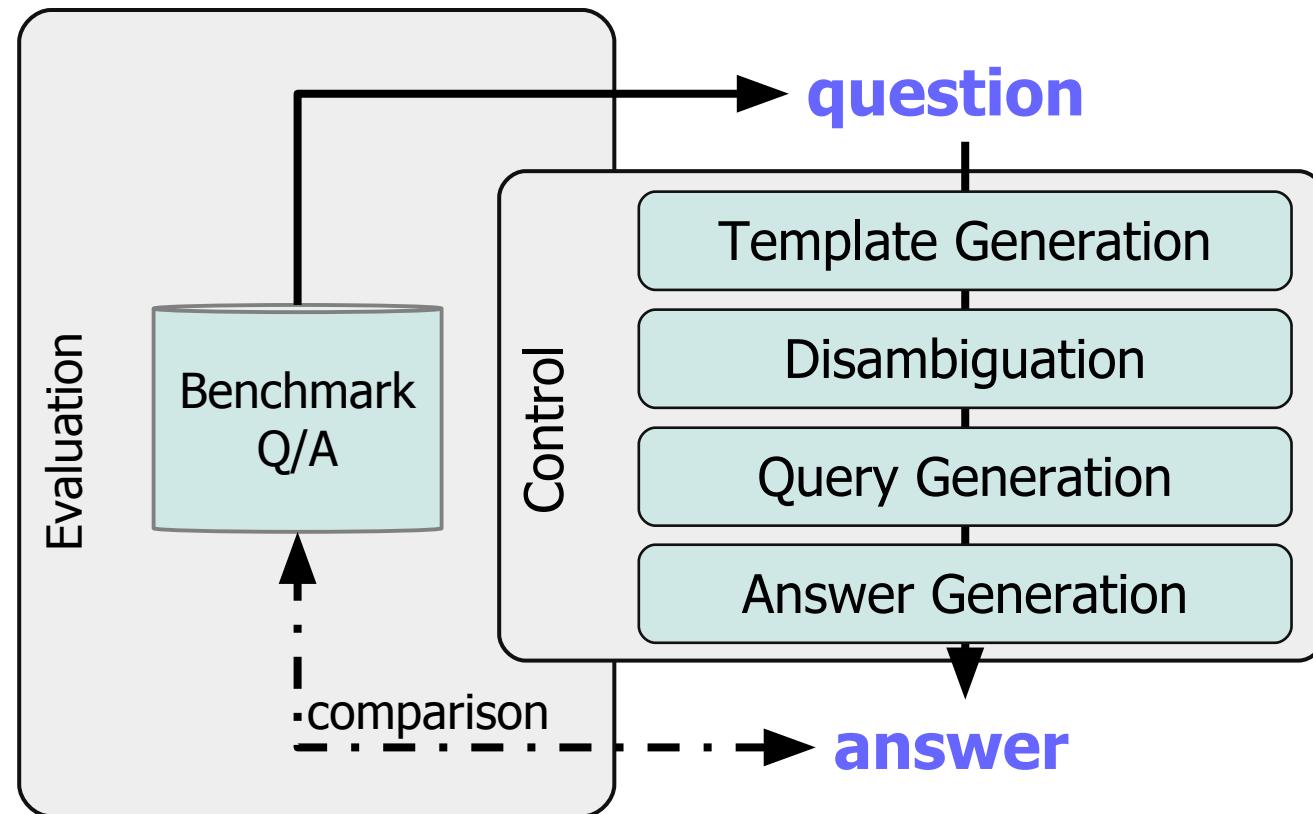
OKBQA framework



OKBQA framework (core)



OKBQA framework (BenchMark)



Components

Template Generation Module (TGM)

- Input: natural language query
- output: structured description of sparql

```
{  
  "language": "en",  
  "string": "which river flows in Seoul?"  
}
```

```
{  
  "query": "SELECT ?v4 WHERE { ?v4 ?v2 ?v6 ; ?v7 ?v3 . } ",  
  "score": "1.0",  
  "slots": [  
    { "s": "v2", "p": "is", "o": "rdf:Property" },  
    { "s": "v2", "p": "verbalization", "o": "flows" },  
    { "s": "v6", "p": "is", "o": "rdf:Resource|rdfs:Literal" },  
    { "s": "v6", "p": "verbalization", "o": "Seoul" },  
    { "s": "v7", "p": "is", "o": "" },  
    { "s": "v3", "p": "is", "o": "rdf:Class" },  
    { "s": "v3", "p": "verbalization", "o": "river" },  
  ]  
}
```

question

Template Generation

Disambiguation

Query Generation

Answer Generation

answer



Disambiguation Module (DM)

Input: SPARQL templates

Output: disambiguation information for each slot

```
"ned": [
  {
    "classes": [
      { "var": "v3", "value": "http://dbpedia.org/ontology/River", "score": 0.57 }
    ],
    "entities": [
      { "var": "v6", "value": "http://dbpedia.org/resource/Seoul", "score": 1 }
    ],
    "literals": [
      { "var": "v6", "value": "Seoul", "score": 1 }
    ],
    "properties": [
      { "var": "v2", "value": "http://dbpedia.org/ontology/city", "score": 0.62 },
      { "var": "v2", "value": "http://dbpedia.org/property/flower", "score": 0.53 }
    ],
    "score": 1
  }
]
```

Query Generation Module (QGM)

Input: SPARQL templates with disambiguation

Output: corresponding SPARQL queries

```
[  
  {  
    "query": "SELECT ?v4 WHERE {  
      ?v4 <http://dbpedia.org/ontology/city> <http://dbpedia.org/resource/Seoul> ;  
      ?v7 <http://dbpedia.org/ontology/River> .  
      FILTER (str(?v7) IN ("http://www.w3.org/1999/02/22-rdf-syntax-ns#type",  
                           "http://www.w3.org/2000/01/rdf-schema#subClassOf"))  
    }",  
    "score": 0.5  
  },  
  ...  
]
```

question

Template Generation

Disambiguation

Query Generation

Answer Generation

answer

Answer Generation Module (AGM)

Input: SPARQL queries

Output: Answers (URIs, Literals)

```
http://dbpedia.org/resource/Han_River_(Korea)
http://dbpedia.org/resource/Cheonggyecheon
http://dbpedia.org/resource/Cheonggyecheon
http://dbpedia.org/resource/Ara_Canal
http://dbpedia.org/resource/Jungnangcheon
http://dbpedia.org/resource/Jungnangcheon
http://dbpedia.org/resource/Han_River_(Korea)
http://dbpedia.org/resource/Yanghwa_Bridge
http://dbpedia.org/resource/Hannam_Bridge
http://dbpedia.org/resource/Banghwa_Bridge
http://dbpedia.org/resource/Jamsu_Bridge
http://dbpedia.org/resource/Hannam_Bridge
http://dbpedia.org/resource/Banghwa_Bridge
http://dbpedia.org/resource/Jamsu_Bridge
http://dbpedia.org/resource/Andrew_Kim_Taegon
http://dbpedia.org/resource/Andrew_Kim_Taegon
http://dbpedia.org/resource/Andrew_Kim_Taegon
...
...
```

question

Template Generation

Disambiguation

Query Generation

Answer Generation

answer



Principles

JSON for I/O

- “a simplified XML”
- Elements
 - Object {}, a.k.a. hash, dictionary, key-value pairs
 - Array []
- Example

```
{  
    "query": "SELECT ?v4 WHERE { ?v4 ?v2 ?v6 ; ?v7 ?v3 . } ",  
    "score": "1.0",  
    "slots": [  
        { "s": "v2", "p": "is", "o": "rdf:Property" },  
        { "s": "v2", "p": "verbalization", "o": "flows" },  
        { "s": "v6", "p": "is", "o": "rdf:Resource|rdfs:Literal" },  
        { "s": "v6", "p": "verbalization", "o": "Seoul" },  
        { "s": "v7", "p": "is", "o": "" },  
        { "s": "v3", "p": "is", "o": "rdf:Class" },  
        { "s": "v3", "p": "verbalization", "o": "river" },  
    ]  
}
```

Components as REST services

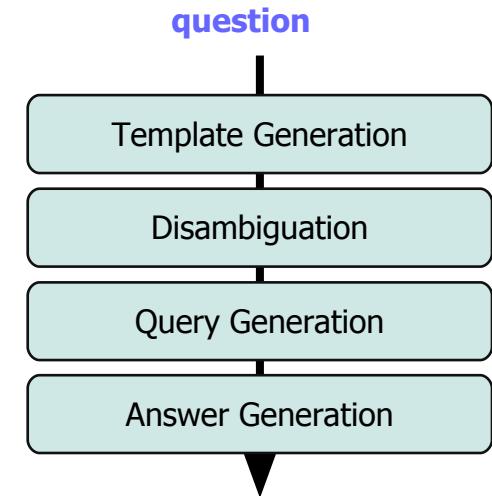
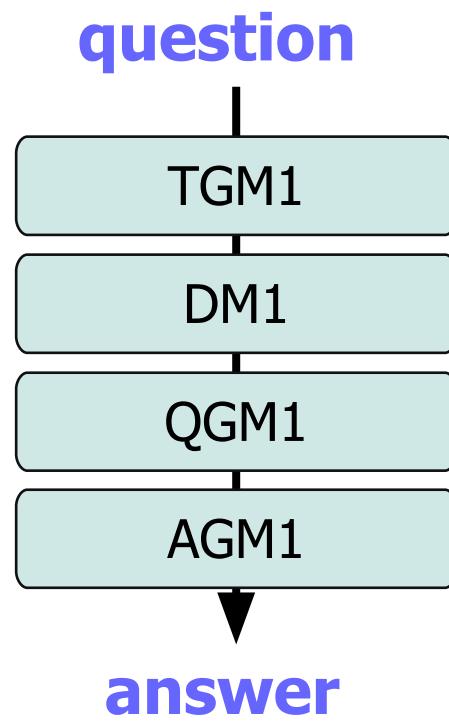
- HTTP POST (or GET) and response
- Example (Post a JSON object in the body of req.)

```
curl -H "Content-Type: application/json"
http://ws.okbqa.org:1555/templategeneration/templator -d '
{
  "string": "Which rivers flow through Seoul?",
  "language": "en"
}
'
```

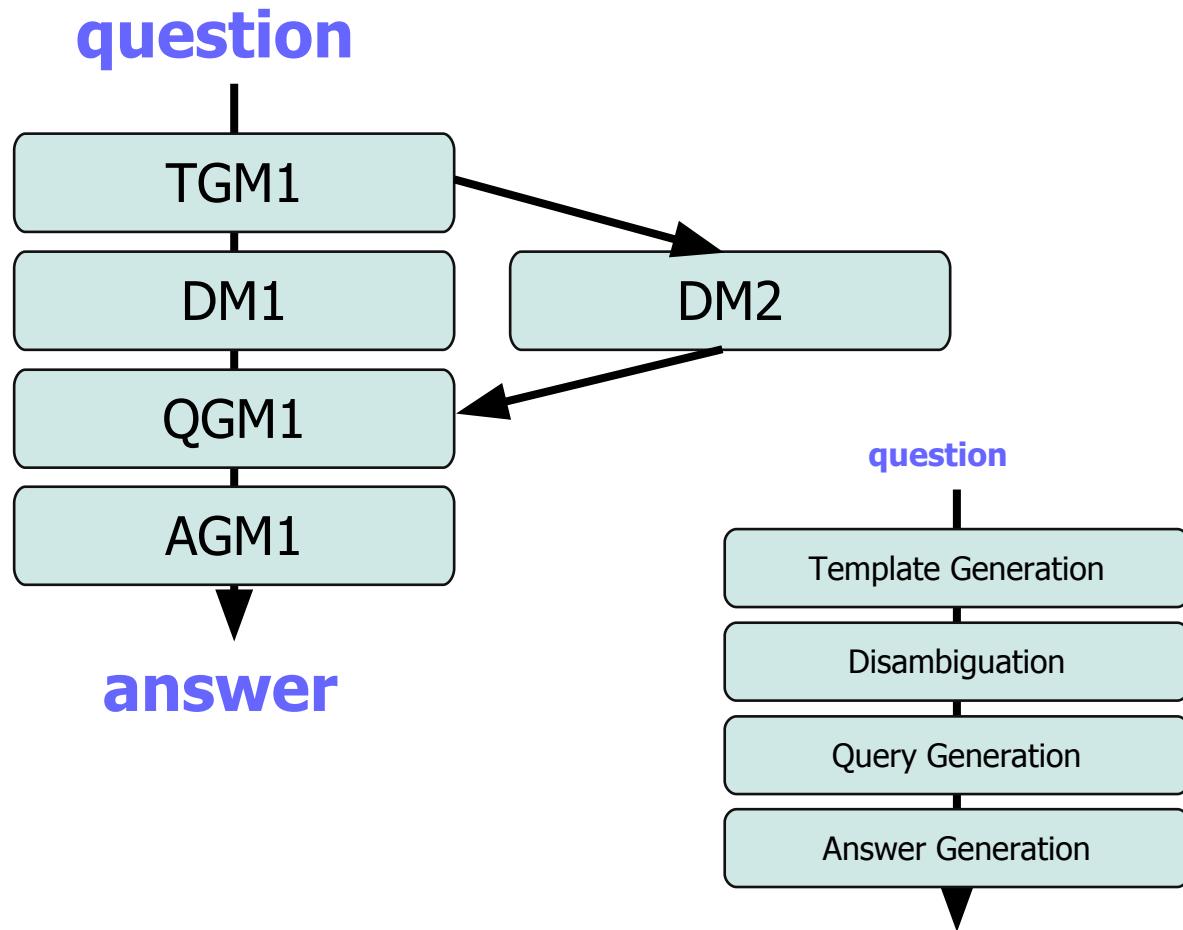
```
[
  {
    "score": "1.0",
    "slots": [
      { "s": "v3", "p": "is", "o": "rdf:Class" },
      ...
    ]
  }
  ...
]
```



Flexible Workflow

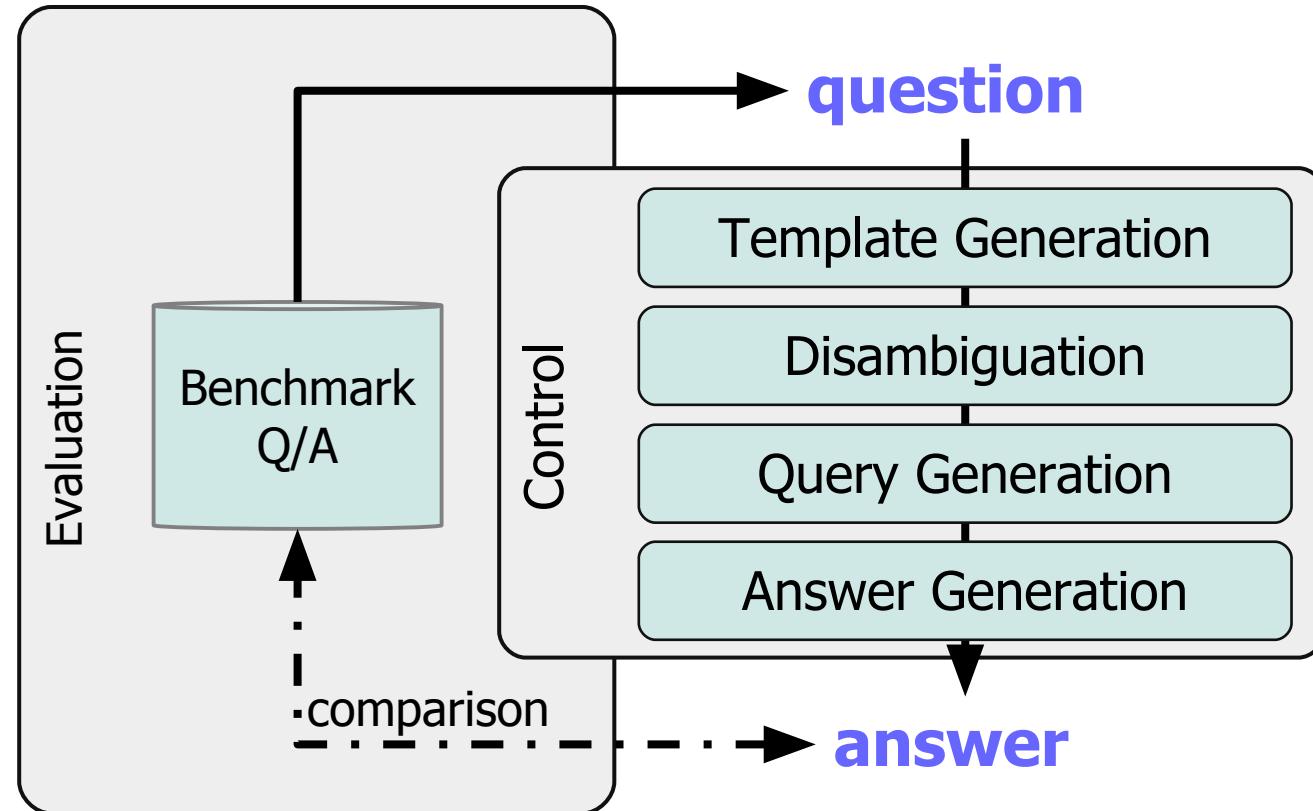


Flexible Workflow



Benchmark Evaluation

- To allow evaluation of improvements



Resources

Documentation

<http://www.okbqa.org>

Repository

- A central venue for collection of OKBQA modules
- URL
 - <http://repository.okbqa.org/>



The screenshot shows the OKBQA Repository homepage. The header features the OKBQA logo (a stylized 'K' inside a circle) and the text "OKBQA Repository" followed by "Open Knowledge Base and Question-Answering". On the right side of the header are "Sign Up" and "Log in" buttons. Below the header is a search bar with placeholder text "Search OKBQA Repository". The main content area is titled "List" and contains a table of components.

List

Title	Category	Description	Maintainer
Sparqlator	Query generation	Description It is a wrapper to call the GraphFinder::sparqlator method, using ...	jindong.kim@gmail.com
Templator	Template generation	Description Templator is a module for dependency-driven SPARQL template ...	cunger@cit-ec.uni-bielefeld.de
AGDISTIS-KO	Disambiguation	# Description Disambiguator uses the AGDISTIS web service to disambiguat...	zakria.ai@gmail.com
SRDF 1.0	Other	Description SRDF is a Korean Open Information Extraction system. It is desi...	nam.sangha@gmail.com
C2K 2.0	Other	C2K is a knowledge acquisition system that extracts triples of DBpedia prope...	jiseong@kaist.ac.kr
Disambiguator	Disambiguation	Description Disambiguator is a module to disambiguate class, properties and...	hahmyg@kaist.ac.kr
TGM JAVA WRAPP...	Template generation	Description It is a wrapper for Java Template Generation Module(TGM). If yo...	nam.sangha@gmail.com
DM JAVA WRAPPER	Disambiguation	Description It is a warpper for Java Disambiguation Module(DM). If you want...	nam.sangha@gmail.com

Controller

- To allow control and execution of the workflow
- URL
 - <http://ws.okbqa.org/~testuser02/>

Conclusions

Conclusions

- OKBQA framework enables
 - Modular development
 - Distributed development
 - Testing different workflows
 - Benchmarking performance of workflows
- Caveat
 - OKBQA framework is optimized for
 - development phase (collaborative development)
 - but not for
 - production phase (fast processing)
- License condition
 - OKBQA modules are recommended to be released under MIT license terms (the most liberal one)



OKBQA-4

TUTORIAL : 16-17
SYMPOSIUM : 18
HACKATHON : 18-21
July 2016, Jeju, Korea



How to develop a template generation module

Christina Unger and Younggyun Hahm

Universität Bielefeld, KAIST



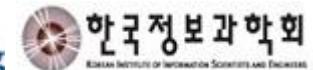
한국연구재단



UNIVERSITÄT LEIPZIG

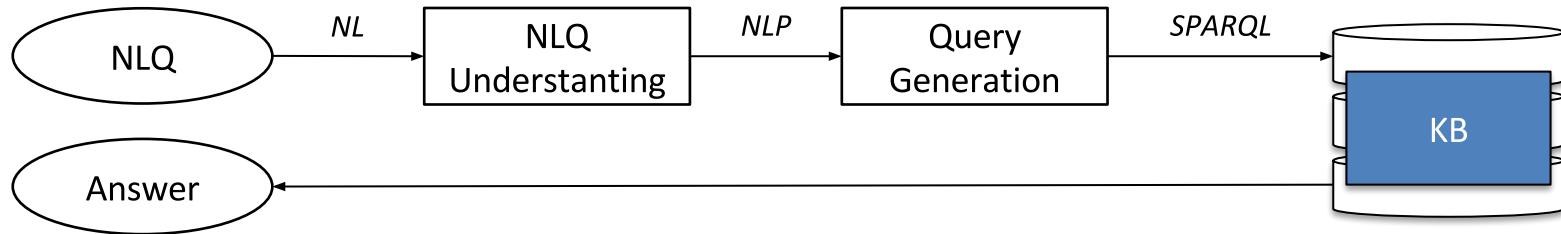


School of Computing



WHAT IS “TEMPLATE GENERATION”?

Background Knowledge: KB-QA and Query Generation



- Query Generation in KB-QA
 - Understands natural language questions and converts it to machine-readable query (e.g. SPARQL)
- NLQ to triple-based representation
 - Who wrote the neverending story?
 - <[person, organization], wrote, Neverending Story>

SPARQL and Quantifier

- (a) Who produced the most films?
- (b) <[person,organization], produced, most films>
- (c)

```
SELECT ?y WHERE {
    ?x rdf:type onto:Film .
    ?x onto:producer ?y .
}
ORDER BY DESC(COUNT(?x)) OFFSET 0 LIMIT 1
```

- (b) → (c)
 - In triple representation (b), specific quantifiers are not faithfully captured

SPARQL and Quantifier

- Solution
 - 1) generate SPARQL template first
 - 2) and then, construct SPARQL template

Who produced the most film?

SPARQL template

```
SELECT ?x WHERE {  
    ?x ?p ?y .  
    ?y rdf:type ?c .  
}  
ORDER BY DESC(COUNT(?y)) LIMIT 1 OFFSET 0
```

SPARQL template

+ slots

```
SELECT ?x WHERE {  
    ?x ?p ?y .  
    ?y rdf:type ?c .  
}  
ORDER BY DESC(COUNT(?y)) LIMIT 1 OFFSET 0
```

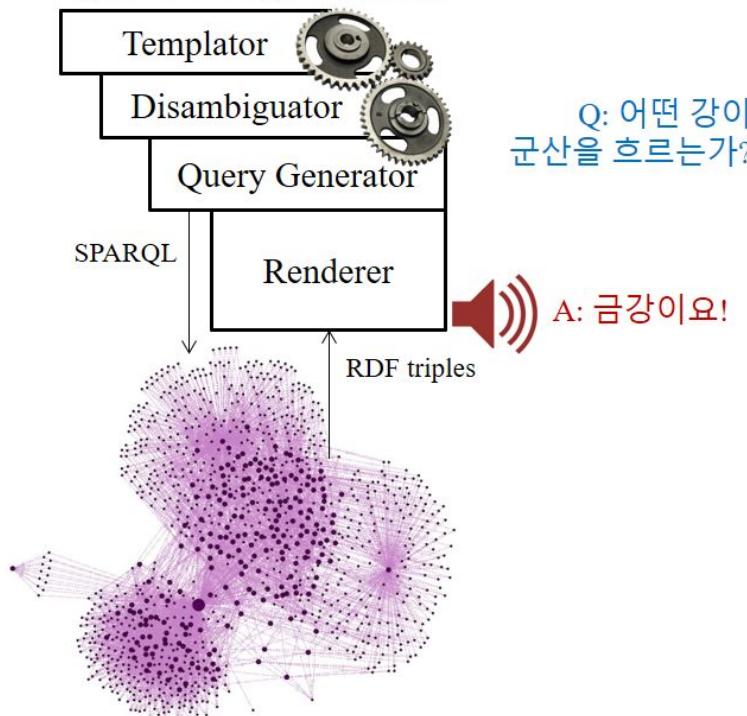
Slots:

- ⟨?c, class, films⟩
- ⟨?p, property, produced⟩



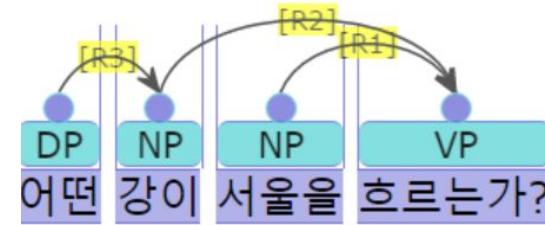
OKBQA and TGM

Open KB & QA Platform



Q: 어떤 강이
군산을 흐르는가?

A: 금강이요!



TGM = Template Generation Module

```
{"question": "어떤 강이 서울을 흐르는가?",  
 "query": "SELECT ?v2 WHERE { ?v2 ?v6 ?v1 . ?v2 ?v3 ?v5 . }",  
 "slots": [ {"s": "v5", "p": "is", "o": "owl:NamedIndividual"},  
           {"s": "v5", "p": "verbalization", "o": "서울"},  
           {"s": "v6", "p": "is", "o": "           {"s": "v1", "p": "is", "o": "owl:Class"},  
           {"s": "v1", "p": "verbalization", "o": "강"},  
           {"s": "v3", "p": "is", "o": "owl:Property"},  
           {"s": "v3", "p": "verbalization", "o": "흐르는가"} ],  
 "score": "1.0" }
```

DM = Disambiguation Module

```
"var": "v1", "value": "http://dbpedia.org/ontology/River"  
"var": "v3", "value": "http://dbpedia.org/ontology/city"
```

QGM = Query Generation Module

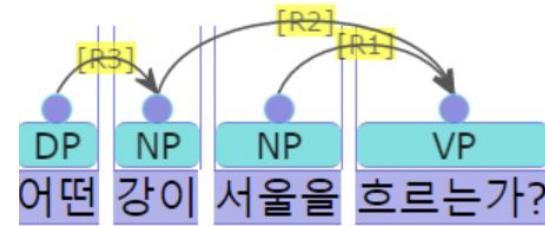
```
SELECT ?v2 WHERE {  
  ?v2 rdf:type http://dbpedia.org/ontology/River> .  
  ?v2 <http://dbpedia.org/ontology/city>  
  <http://dbpedia.org/resource/Seoul> .}
```



OKBQA and TGM

Pattern #n (example)

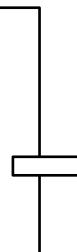
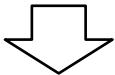
- NP → VP (dependency distant = 1)
 - if VP == HEAD :
 - VP ↳ property
 - if NP == SUBJECT :
 - NP ↳ class
 - if NP is not SUBJECT :
 - NP ↳ Individual
 - if NP is OBJECT :
 - NP ↳ OBJECT_in_RDF_TRIPLE



TGM

```
{"question": "어떤 강이 서울을 흐르는가?",  
"query": "SELECT ?v2 WHERE { ?v2 ?v6 ?v1 . ?v2 ?v3 ?v5 . }",  
"slots": [ {"s": "v5", "p": "is", "o": "owl:NamedIndividual"},  
          {"s": "v5", "p": "verbalization", "o": "서울"},  
          {"s": "v6", "p": "is", "o": ""},  
          {"s": "v1", "p": "is", "o": "owl:Class"},  
          {"s": "v1", "p": "verbalization", "o": "강"},  
          {"s": "v3", "p": "is", "o": "owl:Property"},  
          {"s": "v3", "p": "verbalization", "o": "흐르는가"} ],  
"score": "1.0" }
```

- 흐르는가
 - ↳ owl:Property
- 서울
 - ↳ owl:NamedIndividual
- 강
 - ↳ owl:Class



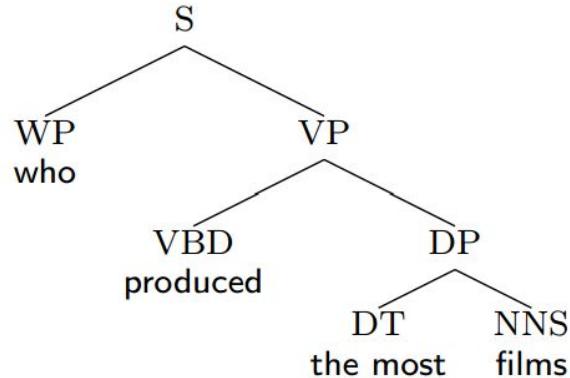
```
SELECT ?y WHERE {  
    ?x 흐르는가 서울 .  
    ?y rdf:type 강 .  
}
```



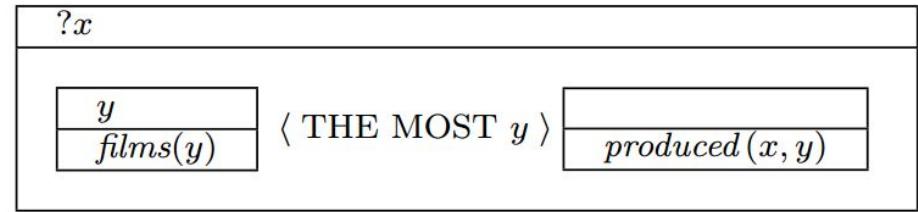
How TGM work

NLQ

Who produced the most films?



syntactic structure



semantic representation

* Christina Unger et al., Template-based Question Answering over RDF data, 2012, Proceedings of the 21st international conference on World Wide Web. ACM, 2012.

How TGM work

NLQ

Who produced the **most** films?



SPARQL TEMPLATE

```
SELECT ?x WHERE {  
    ?x ?p ?y  
    ?y rdf:type ?c  
}  
ORDER BY DESC(COUNT(?x)) OFFSET 0 LIMIT 1
```

* Christina Unger et al., Template-based Question Answering over RDF data, 2012, Proceedings of the 21st international conference on World Wide Web. ACM, 2012.

How TGM work

NLQ

Who **produced** the most **films**?



SPARQL TEMPLATE

```
SELECT ?x WHERE {  
    ?x ?p ?y .  
    ?y rdf:type ?c .  
}  
ORDER BY DESC(COUNT(?X)) OFFSET 0 LIMIT 1
```

SLOTS

<?c, class, **films**>
<?p property, **produced**>

* Christina Unger et al., Template-based Question Answering over RDF data, 2012, Proceedings of the 21st international conference on World Wide Web. ACM, 2012.

How TGM work

SPARQL TEMPLATE

```
SELECT ?x WHERE {  
    ?x ?p ?y .  
    ?y rdf:type ?c .  
}  
ORDER BY DESC(COUNT(?X)) OFFSET 0 LIMIT 1
```

SLOTS

<?c, class, films>
<?p property, produced>



SPARQL QUERY

```
SELECT ?y WHERE {  
    ?x produced ?y .  
    ?y rdf:type films .  
}  
ORDER BY DESC(COUNT(?X)) OFFSET 0 LIMIT 1
```

* Christina Unger et al., Template-based Question Answering over RDF data, 2012, Proceedings of the 21st international conference on World Wide Web. ACM, 2012.



OKBQA-4

TUTORIAL : 16-17
SYMPOSIUM : 18
HACKATHON : 18-21
July 2016, Jeju, Korea



How to develop a disambiguation module

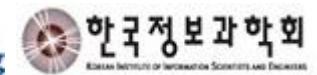
Jeonguk Kim

Master's student @ KAIST

Contact: prismriver@kaist.ac.kr



UNIVERSITAT LEIPZIG



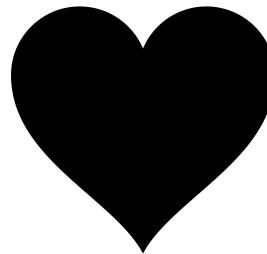
Background Knowledge: Definition of Entity

- Entity
 - something that exists in itself separately from other thing



Background Knowledge: Definition of Entity

- Entity
 - something that exists in itself separately from other thing



- may be a set, abstract noun, unique noun, a person, and so on.



Background Knowledge: Definition of Entity

- Entity
 - entity can be represented as text, picture, and so on
 - requires a way to group & distinguish entities



Background Knowledge: Definition of Entity

- Entity
 - entity can be represented as text, picture, and so on
 - requires a way to group & distinguish entities
 - URI of knowledge base! (e.g. Wikipedia)



박근혜

박근혜

위키백과, 우리 모두의 백과사전.

박근혜(朴槿惠)^[8], 1952년 2월 2일 ~)는 대한민국의 제18대 대통령이다.^[9] [10]

제5·6·7·8·9대 대통령 박정희와 육영수 사이의 장녀이다. 본관은 고령이다. 1963년 아버지 박정희의 대통령 취임 이후 청와대에서 성장하였다. 1970년에 서강대학교에 입학하여 1974년에 졸업한 이후 프랑스 그로노를대학교로 유학을 떠났으나 1974년 모친 사후 귀국하여 1979년 10.26 사건 이전까지 사실상 영부인 직무를 대행하였다. 아버지가 서거한 후 청와대에서 나와 육영재단 이사장과 정수장학회의 이사장 등을 지냈다.

1998년 정계에 입문하여 4.2 재보궐선거에서 당선되었고 이후 제19대까지 5선 국회의원을 지냈다. 2004년부터 2006년까지 한나라당의 대표최고위원을 역임하였으며, 2007년 한나라당의 제17대 대선 후보 경선에 출마하였지만 이명박에게 석패하였다. 이후 이명박 정부 기간 동안 한나라당 내 비주류계를 이끌었다.

2011년 12월부터는 새누리당 비상대책위원장과 함께 당 혁신 작업을 지휘하였고 2012년 제19대 총선을 승리로 이끌었다. 이후 새누리당의 대통령 선거 후보로 선출되어 2012년 12월 19일 실시된 제18대 대선에서 51.6%의 득표율로 민주통합당의 문재인 후보를 누르고 당선되었다.^[11]

2013년 2월 25일 대한민국의 제18대 대통령으로 취임하였으며, 이로써 대한민국 최초의 여성 대통령이자 1987년 대한민국 헌법 개정 이후 최초의 과반 득표 대통령, 최초의 이공계 출신 대통령, 최초의 독신 대통령, 부녀 대통령으로 기록되었다.^[12]



Background Knowledge: Entity Disambiguation

- Entity disambiguation
 - finding a correct URI from the entity among multiple candidates

Background Knowledge: Entity Disambiguation

- Entity disambiguation
 - finding a correct URI from the entity among multiple candidates

고종은 조선의 26대 임금이다.

https://ko.wikipedia.org/wiki/고려_고종

https://ko.wikipedia.org/wiki/대한제국_고종

https://ko.wikipedia.org/wiki/리_고종

https://ko.wikipedia.org/wiki/당_고종

Background Knowledge: Entity Disambiguation

- Entity disambiguation
 - finding a correct URI from the entity among multiple candidates

고종은 조선의 26대 임금이다.

https://ko.wikipedia.org/wiki/고려_고종

https://ko.wikipedia.org/wiki/대한제국_고종

https://ko.wikipedia.org/wiki/리_고종

https://ko.wikipedia.org/wiki/당_고종



Background Knowledge: Relation Disambiguation

- Relation Disambiguation
 - finding a correct property from the given ontology



Background Knowledge: Relation Disambiguation

- Relation Disambiguation
 - finding a correct property from the given ontology
 - “이순신은 노량해전에서 사망하였다.”

| s | p | o |
|---------|----|-----------|
| 이순신 | 사망 | 노량해전 |
| dbr:이순신 | 사망 | dbr:노량_해전 |



?

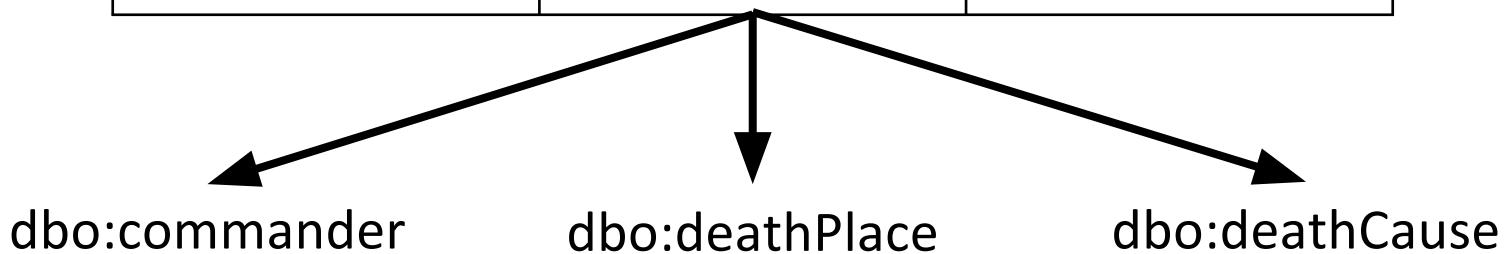
Background Knowledge: Relation Disambiguation

- Relation Disambiguation
 - Finding relation candidates in KB
 - Semantic-based Prediction
 - dbr:이순신 rdf:type dbo:MilitaryPerson
 - dbr: 노량_해전 rdf:type dbo:MilitaryConflict
 - dbr:이순신 **dbo:commander** dbr: 노량_해전
 - Lexicon-based Prediction
 - 사망 → death
- dbo:deathPlace / dbo:deathCause / dbo:deathDate**
prop-ko:사망지 / prop-ko:사망일

Background Knowledge: Relation Disambiguation

- Relation Disambiguation
 - finding a correct property from the given ontology
 - “이순신은 노량해전에서 사망하였다.”

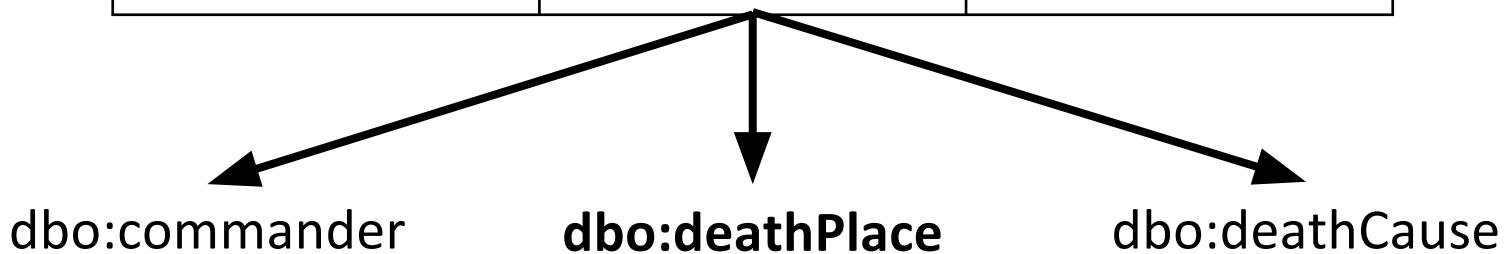
| s | p | o |
|---------|----|-----------|
| 이순신 | 사망 | 노량해전 |
| dbr:이순신 | 사망 | dbr:노량_해전 |



Background Knowledge: Relation Disambiguation

- Relation Disambiguation
 - finding a correct property from the given ontology
 - “이순신은 노량해전에서 사망하였다.”

| s | p | o |
|---------|----|-----------|
| 이순신 | 사망 | 노량해전 |
| dbr:이순신 | 사망 | dbr:노량_해전 |



Background Knowledge: OKBQA Disambiguation Module

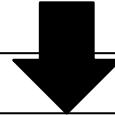
- Disambiguation Module
 - **disambiguates entity / disambiguates relation**

```
{ "question": "어떤 강이 서울을 흐르는가?",  
  "query": "SELECT ?v2 WHERE  
           { ?v2 ?v6 ?v1 . ?v2 ?v3 ?v5 . }",  
  "slots": [ {"s": "v5", "p": "is", "o": "owl:NamedIndividual"},  
            {"s": "v5", "p": "verbalization", "o": "서울"},  
            {"s": "v6", "p": "is", "o": "<http://lodqa.org/vocabulary/sort_of>"},  
            {"s": "v1", "p": "is", "o": "owl:Class"},  
            {"s": "v1", "p": "verbalization", "o": "강"},  
            {"s": "v3", "p": "is", "o": "owl:Property"},  
            {"s": "v3", "p": "verbalization", "o": "흐르는가"} ],  
  "score": "1.0" }
```

Background Knowledge: OKBQA Disambiguation Module

- Disambiguation Module
 - disambiguates entity / disambiguates relation

```
{ "question": "어떤 강이 서울을 흐르는가?",  
  "query": "SELECT ?v2 WHERE  
           { ?v2 ?v6 ?v1 . ?v2 ?v3 ?v5 . }",  
  "slots": [ {"s": "v5", "p": "is", "o": "owl:NamedIndividual"},  
            {"s": "v5", "p": "verbalization", "o": "서울"},  
            {"s": "v6", "p": "is", "o": "<http://lodqa.org/vocabulary/sort_of>"},  
            {"s": "v1", "p": "is", "o": "owl:Class"},  
            {"s": "v1", "p": "verbalization", "o": "강"},  
            {"s": "v3", "p": "is", "o": "owl:Property"},  
            {"s": "v3", "p": "verbalization", "o": "흐르는가"} ],  
  "score": "1.0" }
```



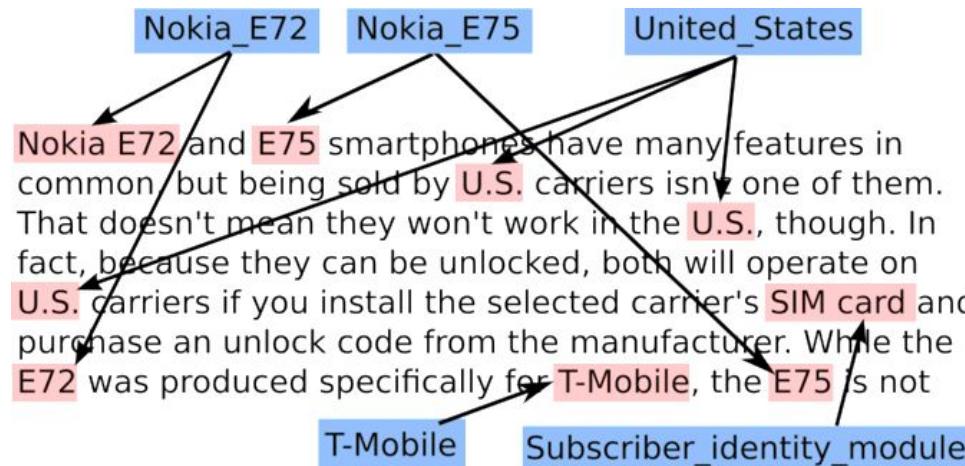
```
[ {"var": "v1", "value": "http://dbpedia.org/ontology/River"},  
  {"var": "v3", "value": "http://dbpedia.org/ontology/city"},  
  {"var": "v5", "value": "http://ko.dbpedia.org/resource/서울특별시"},  
  {"var": "v6", "value": "rdf:type"} ]
```



2. OVERVIEW: KOREAN ENTITY LINKING PROGRAM

Background Knowledge: Korean Entity Linking Program

- Entity linking is the task of **determining the identity of entities** mentioned in text.
 - Named Entity Detection(NED): Detecting entity ‘mentions’.
 - Entity Disambiguation: Linking each mention with an appropriate entity.



Background Knowledge: Korean Entity Linking Program

- Named Entity Detection
 - Find all cases that consecutive morphemes correspond to one of the retrieved labels.

조선 세종

조선

위키백과, 우리 모두의 백과사전.

이 문서의 내용은 출처가 분명하지 않습니다.
지금 바로 이 문서를 편집하여, 참고하신 문헌이나 신뢰할 수 있는 출
되지 않은 내용은 삭제될 수도 있습니다. 내용에 대한 의견은 토론 문

조선은 고조선, 조선민주주의인민공화국 또는 한반도(조선반도) 전체를 뜻하기도 합니다. 그
십시오.

이 문서는 1392년에 이성계가 건국한 국가에 관한 것입니다. 기원전에 단군이 건국한 국가에
조선(朝鮮, 1392년~1897년)은 한반도와 그 부속 도서를 505년간 통치하였던 왕조이다. 공
식 명칭으로 조선국(朝鮮國)이라 하였으며, 때로 대조선국(大朝鮮國)이라는 명칭을 어보
(御寶), 국서(國書) 등에 사용하였다.^[1] 흔히 조선왕조(朝鮮王朝)라고 부르며, 일제 강점기
이후에 이씨조선(李氏朝鮮)이라고 부르기도 한다.^{[2][3][4]} 1897년부터 고종이 대한제국을
선포하여 조선은 역사 속으로 사라지게 되었다.

고려 말기 신진사대부의 지원을 받은 무관 이성계에 의해 건국되었으며 이후 500여 년 동
안 존속됐다. 조선은 유교에 의한 통치 이념을 기본으로 임금과 신하에 의한 치를 중요시
했다.

Background Knowledge: Korean Entity Linking Program

- Named Entity Detection
 - Find all cases that consecutive morphemes correspond to one of the retrieved labels.

조선 세종

조선 세종

위키백과, 우리 모두의 백과사전.



세종은 여기로 연결됩니다. 다른 뜻에 대해서는 세종 (동음이의) 문서를 참조하십시오.

조선 세종(朝鮮 世宗, 1397년 5월 7일^[1] (음력 4월 10일) ~ 1450년 3월 30일 (음력 2월 17일), 재위 1418년 ~ 1450년)은 조선의 제4대 왕, 언어학자이다. 성(姓)은 이(李), 휘(諱)는 도(道), 본관(本貫)은 전주(全州), 자(字)는 원정(元正), 아명은 막동(莫同)이다. 세종은 묘호(廟號)이며, 시호(諡號)는 영문예무인성명효대왕(英文睿武仁聖明孝大王)이고, 명에서 받은 시호는 장현(莊憲)이다. 존시를 합치면 세종장현영문예무인성명효대왕이 된다. 태종(太宗)과 원경왕후(元敬王后)의 셋째 아들이며, 비는 청천부원군(青川府院君) 심온(沈溫)의 딸 소현왕후 심씨(昭憲王后 沈氏)이며, 조선의 왕 중에서 왕세자에게 양위를 하지 않고 봉어한 최초의 왕이다.^{[2][3]}



Background Knowledge: Korean Entity Linking Program

- Named Entity Detection
 - Find all cases that consecutive morphemes correspond to one of the retrieved labels.

조선 세종



조선 세종

위키백과, 우리 모두의 백과사전.

 세종은 여기로 연결됩니다. 다른 뜻에 대해서는 세종 (동음이의) 문서를 참조하십시오.

조선 세종(朝鮮 世宗, 1397년 5월 7일^[1] (음력 4월 10일) ~ 1450년 3월 30일 (음력 2월 17일), 재위 1418년 ~ 1450년)은 조선의 제4대 왕, 언어학자이다. 성(姓)은 이(李), 휘(諱)는 도(道), 본관(本貫)은 전주(全州), 자(字)는 원정(元正), 아명은 막동(莫同)이다. 세종은 묘호(廟號)이며, 시호(諡號)는 영문예무인성명효대왕(英文睿武仁聖明孝大王)이고, 명에서 받은 시호는 장현(莊憲)이다. 존시를 합치면 세종장현영문예무인성명효대왕이 된다. 태종(太宗)과 원경왕후(元敬王后)의 셋째 아들이며, 비는 청천부원군(青川府院君) 심온(沈溫)의 딸 소현왕후 심씨(昭憲王后 沈氏)이며, 조선의 왕 중에서 왕세자에게 양위를 하지 않고 봉어한 최초의 왕이다.^{[2][3]}

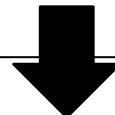


3. EXAMPLE AND DEMO: DISAMBIGUATION MODULE IN OKBQA

Example: Disambiguation Module in OKBQA

- Disambiguation Module
 - disambiguates entity / finds corresponding property

```
{ "question": "어떤 강이 서울을 흐르는가?",  
  "query": "SELECT ?v2 WHERE  
    { ?v2 ?v6 ?v1 . ?v2 ?v3 ?v5 . }",  
  "slots": [ {"s": "v5", "p": "is", "o": "owl:NamedIndividual"},  
            {"s": "v5", "p": "verbalization", "o": "서울"},  
            {"s": "v6", "p": "is", "o": "<http://lodqa.org/vocabulary/sort_of>"},  
            {"s": "v1", "p": "is", "o": "owl:Class"},  
            {"s": "v1", "p": "verbalization", "o": "강"},  
            {"s": "v3", "p": "is", "o": "owl:Property"},  
            {"s": "v3", "p": "verbalization", "o": "흐르는가"} ],  
  "score": "1.0" }
```



```
[ {"var": "v1", "value": "http://dbpedia.org/ontology/River"},  
  {"var": "v3", "value": "http://dbpedia.org/ontology/city"},  
  {"var": "v5", "value": "http://ko.dbpedia.org/resource/서울특별시"},  
  {"var": "v6", "value": "rdf:type"} ]
```



Example: Disambiguation Module in OKBQA

- Entity Disambiguation
 - Input : [{"s": "v5", "p": "is", "o": "owl:NamedIndividual"}, {"s": "v5", "p": "verbalization", "o": "서울시"} ...]
 - Output : {"var": "v5", "value": dbr:서울특별시 }
- Utilizing entity linking module can solve the issue!

Example: Disambiguation Module in OKBQA

- Entity Disambiguation
 - Input : [{"s": "v5", "p": "is", "o": "owl:NamedIndividual"}, {"s": "v5", "p": "verbalization", "o": "서울시"} ...]
 - Output : {"var": "v5", "value": dbr:서울특별시 }
- Entity linking module detects
 - <https://ko.wikipedia.org/resource/> 서울특별시
 - <https://ko.wikipedia.org/resource/> 시_(행정_구역)

Example: Disambiguation Module in OKBQA

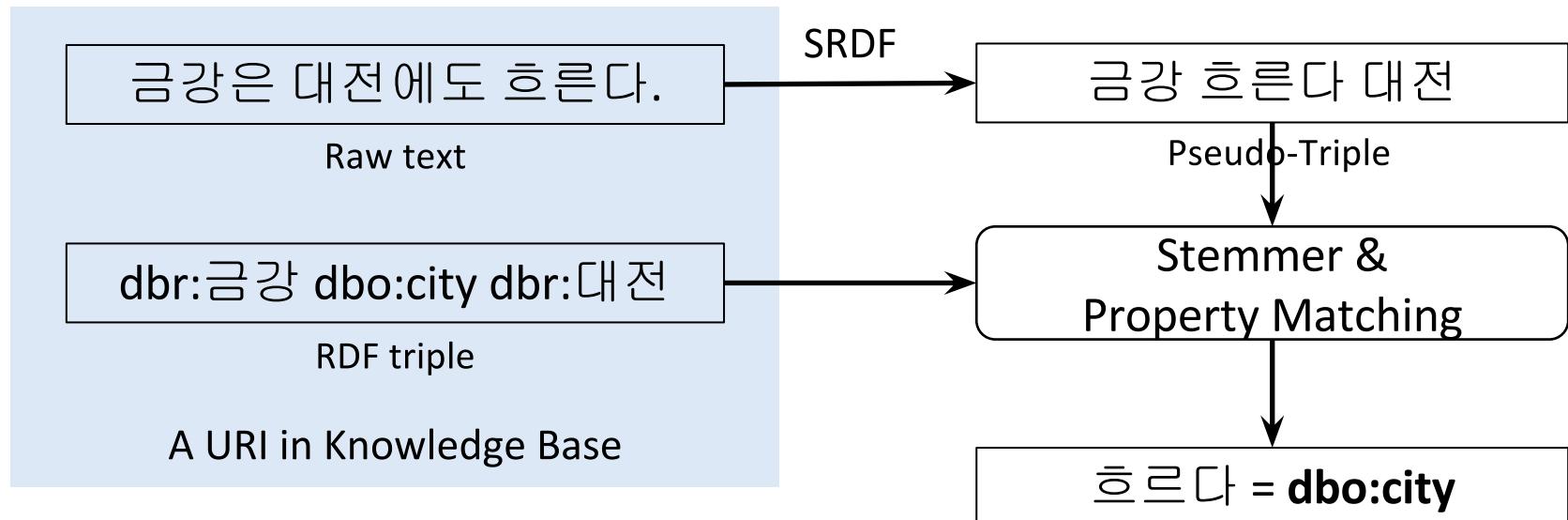
- Entity Disambiguation
 - Input : [{"s": "v5", "p": "is", "o": "owl:NamedIndividual"}, {"s": "v5", "p": "verbalization", "o": "서울시"} ...]
 - Output : {"var": "v5", "value": dbr:서울특별시 }
- Entity linking module detects
 - <https://ko.wikipedia.org/resource/서울특별시>
 - [https://ko.wikipedia.org/resource/시_\(행정_구역\)](https://ko.wikipedia.org/resource/시_(행정_구역))
- Select entity that embraces the other

Example: Disambiguation Module in OKBQA

- Relation disambiguation
 - Input : [{"s": "v3", "p": "is", "o": "owl:Property"}, {"s": "v3", "p": "verbalization", "o": "흐르는가"} ...]
 - Output : {"var": "v3", "value": "http://dbpedia.org/ontology/city"}
- Use triple & raw text in knowledge base to match appropriate property of the given word

Example: Disambiguation Module in OKBQA

- Relation disambiguation
 - Input : [{"s": "v3", "p": "is", "o": "owl:Property"}, {"s": "v3", "p": "verbalization", "o": "흐르는가"} ...]
 - Output : {"var": "v3", "value": "http://dbpedia.org/ontology/city"}





OKBQA-4

TUTORIAL : 16-17
SYMPORIUM : 18
HACKATHON : 18-21
July 2016, Jeju, Korea



Query Generation

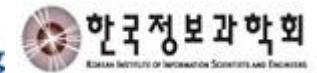
Jin-Dong Kim

Database Center for Life Science (DBCLS)

Contact: jdkim@dbcls.rois.ac.jp



UNIVERSITAT LEIPZIG



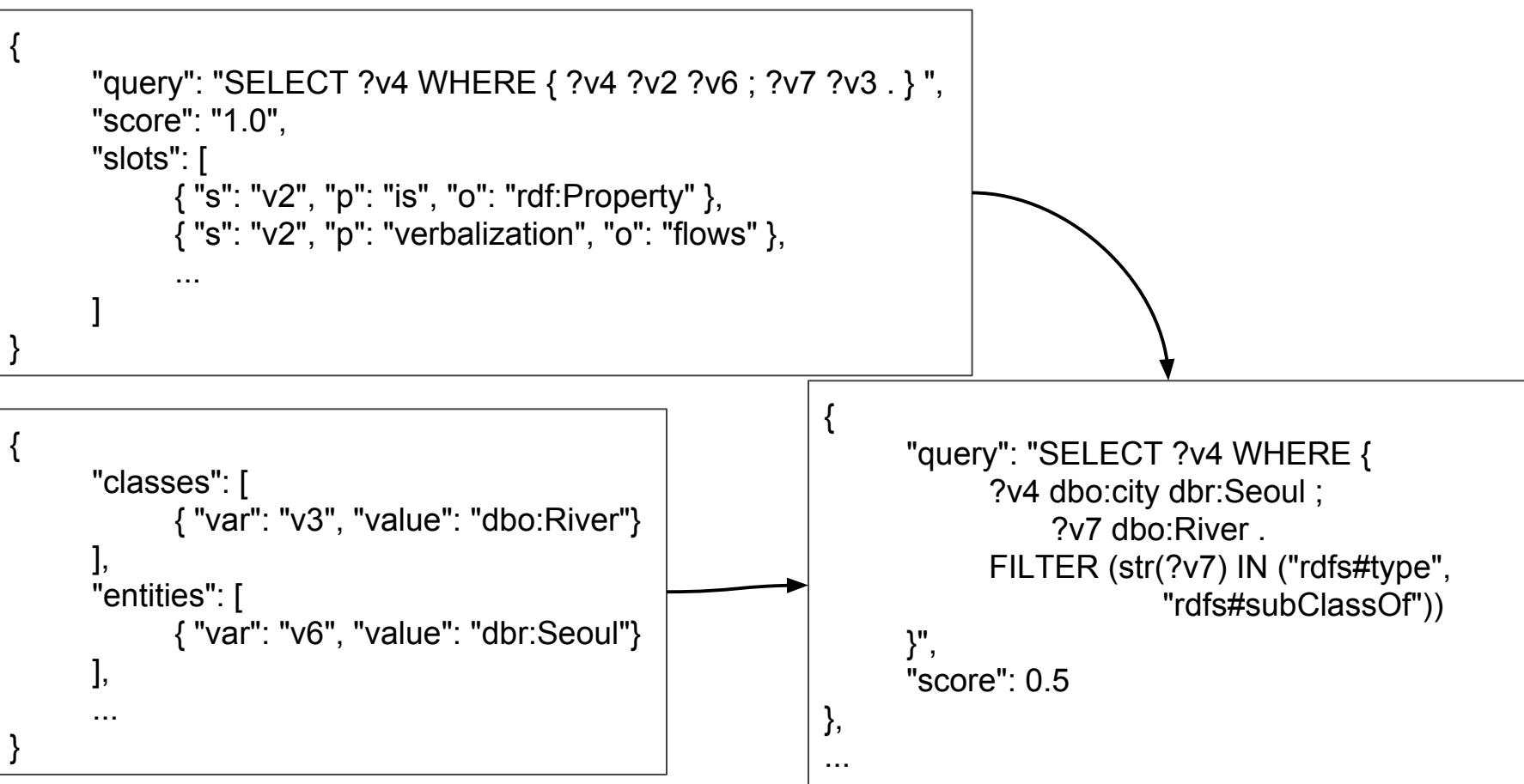
Query Generation

- Objective
 - To generate SPARQL queries which are ready to be executed
- Given
 - SPARQL templates
 - disambiguation information
- Problem to be solved
 - representation variation



Query Generation

- Input: Template + Disambiguation
- output: SPARQL query



Query Generation

```
SELECT ?v4  
WHERE {  
    ?v4 ?v2 ?v6 ;  
    ?v7 ?v3 .  
}
```

```
"classes": [  
    { "var": "v3", "value": "dbo:River"}  
],  
"entities": [  
    { "var": "v6", "value": "dbr:Seoul"}  
],  
"literals": [  
    { "var": "v6", "value": "Seoul"}  
],  
"properties": [  
    { "var": "v2", "value": "dbo:city"},  
    { "var": "v2", "value": "dbp:flower"}  
]
```

```
SELECT ?v4  
WHERE {  
    ?v4 dbo:city dbr:Seoul ;  
    a dbo:River .  
}
```



Answer Generation

To submit SPARQL queries and collect answers

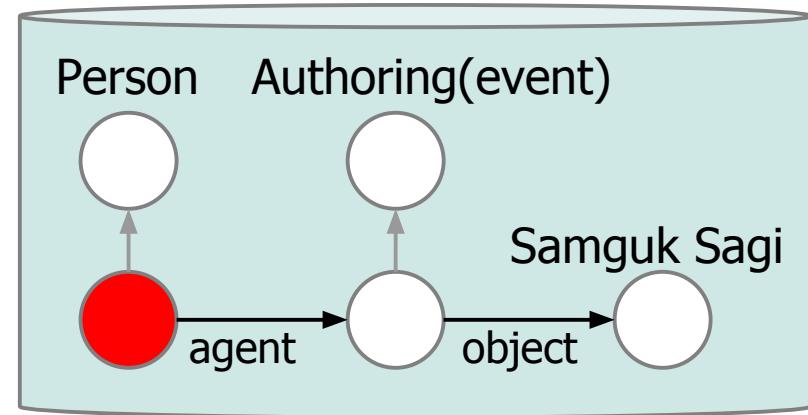
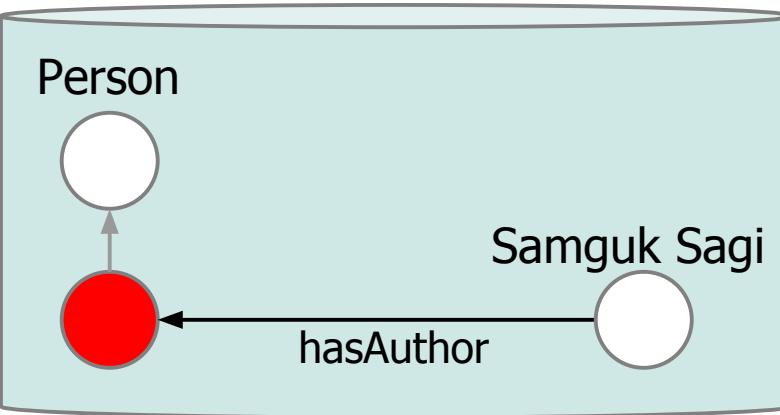
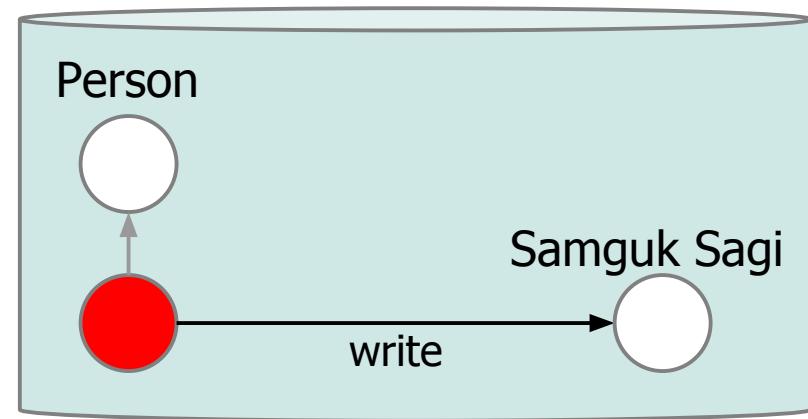
DBpedia SPARQL endpoint

- <http://dbpedia.org/sparql>
- <http://143.248.135.20:45103/sparql>
 - Graph URI: <http://en.dbpedia2015-10.kaist.ac.kr>

Problem

- Template may be different from actual data
 - model representation vs. data representation

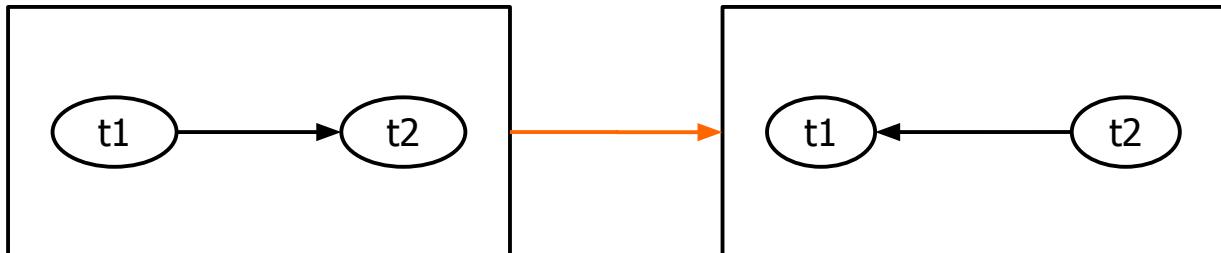
Who wrote "Samguk Sagi"?



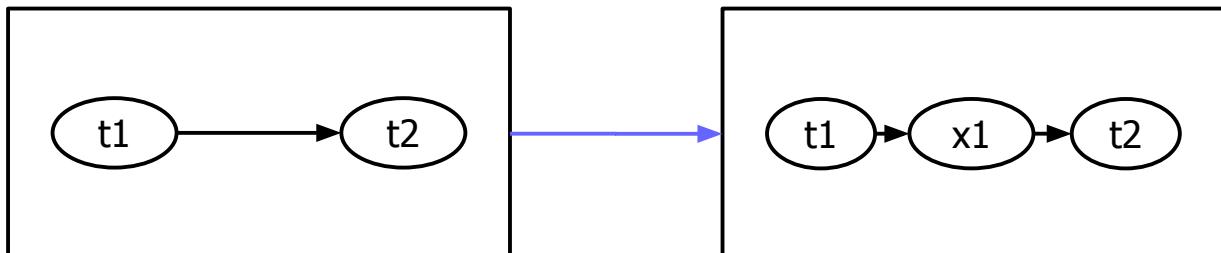
GraphFinder

Graph Variation Operations (GVO)

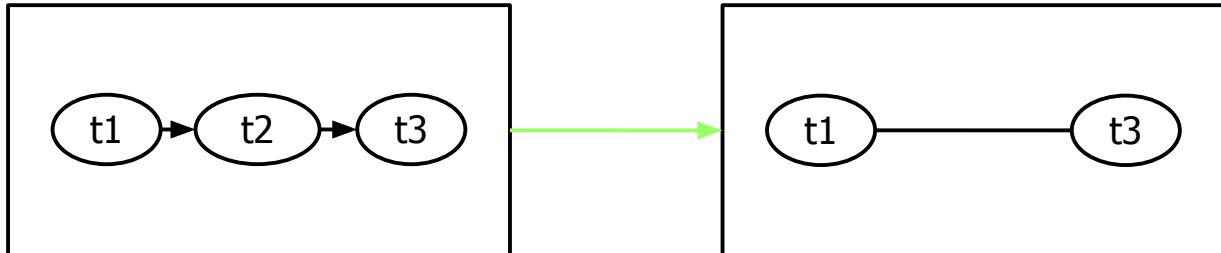
① inversion



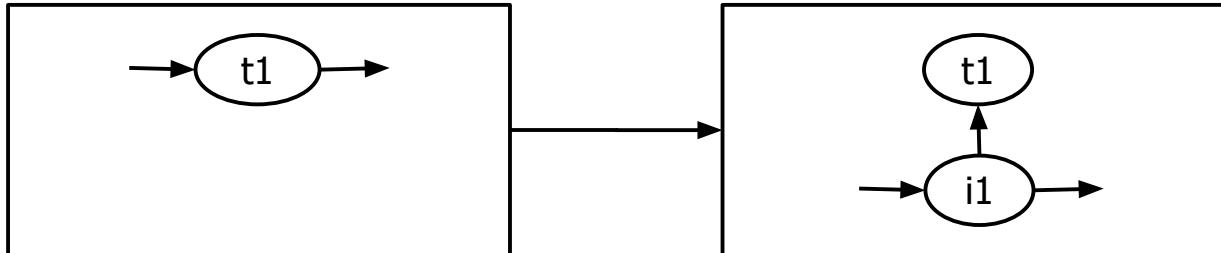
② split



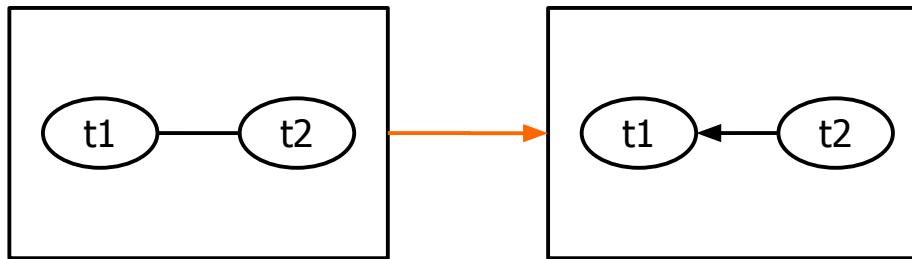
③ join



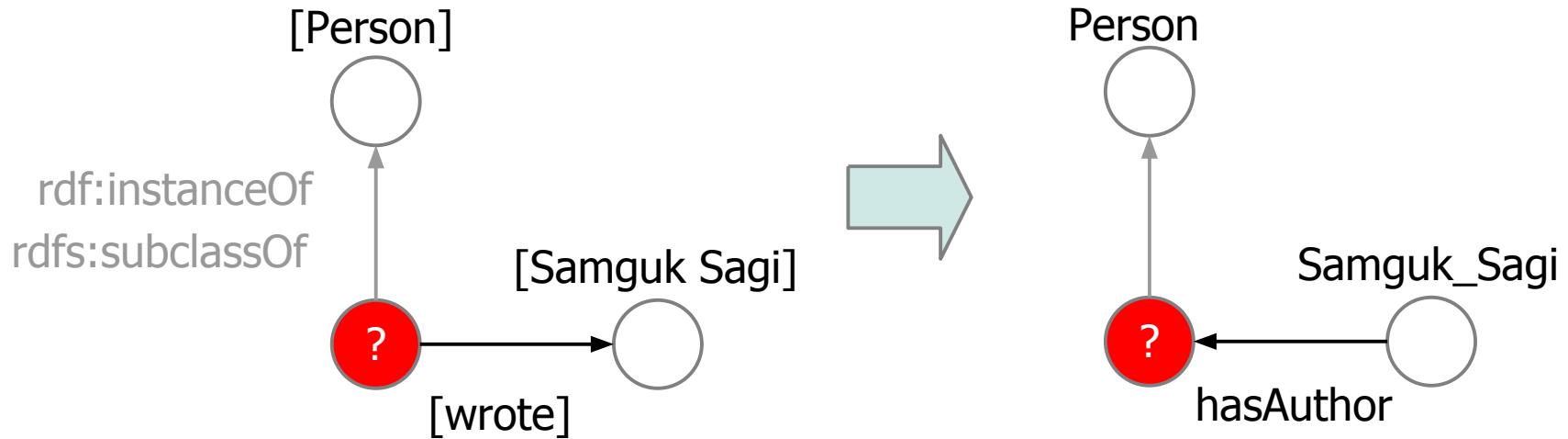
④ instantiation



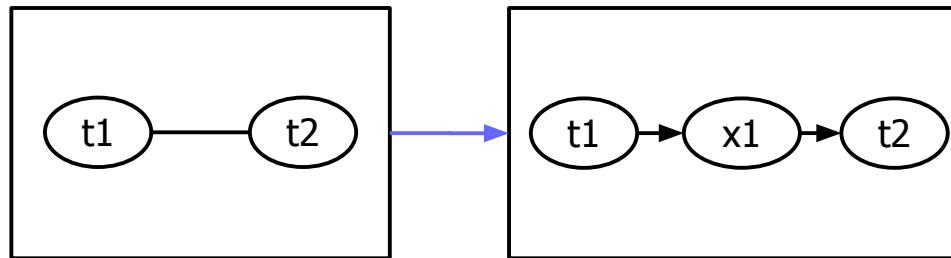
SVO 1. Inversion



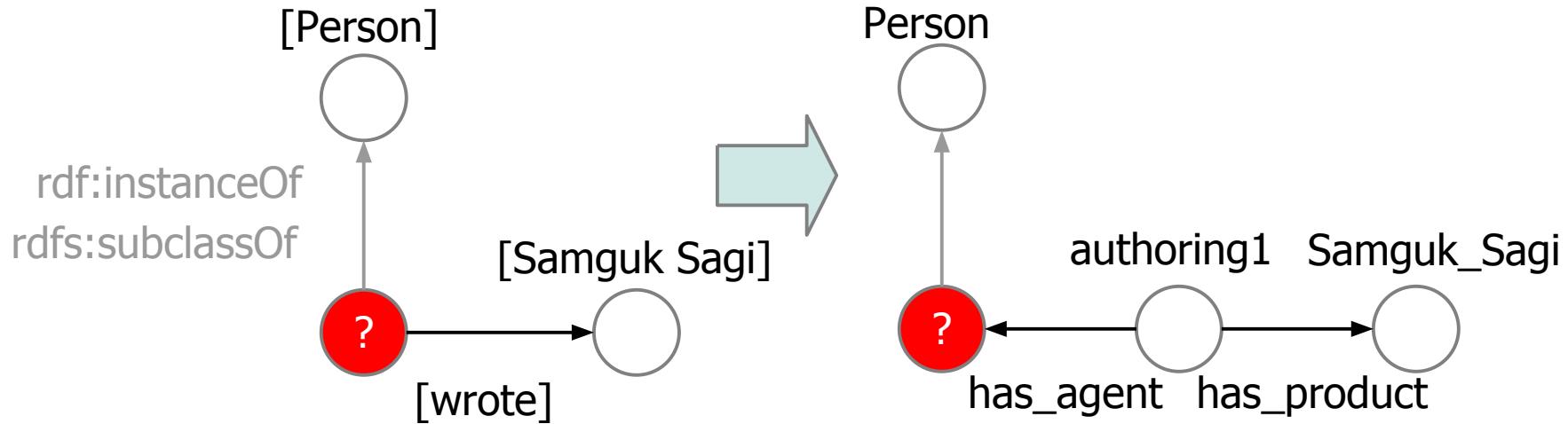
Who wrote "Samguk Sagi"?



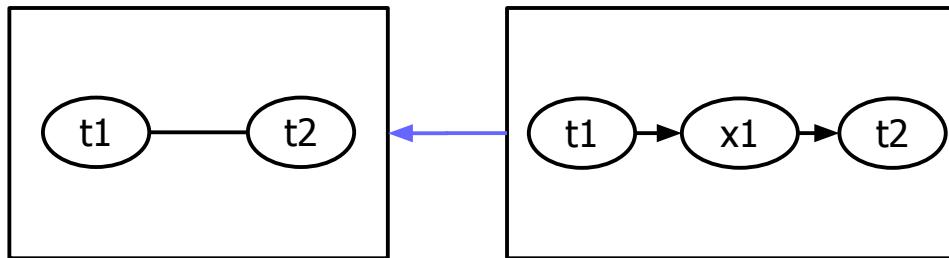
SVO 2. Split



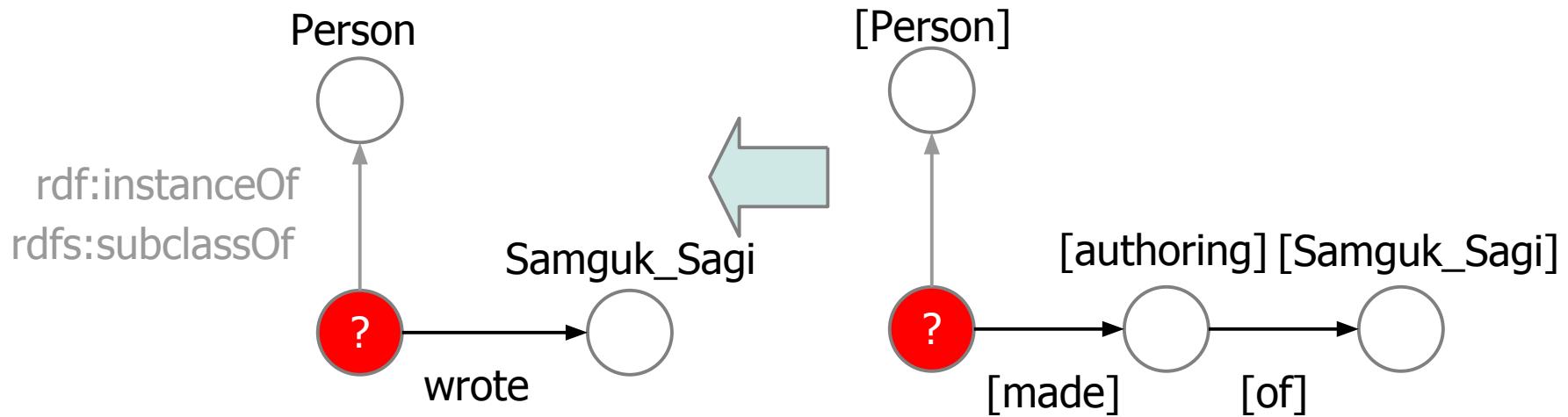
Who wrote "Samguk Sagi"?



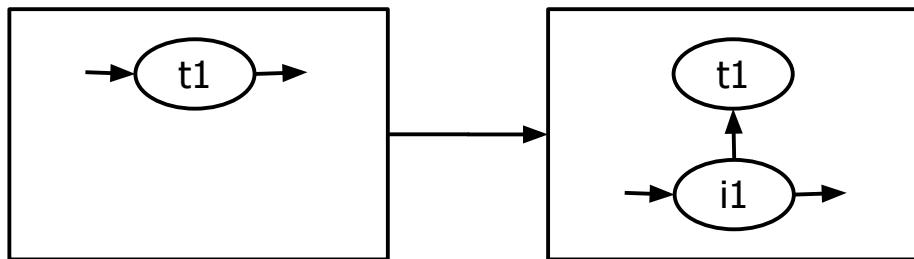
SVO 3. Join



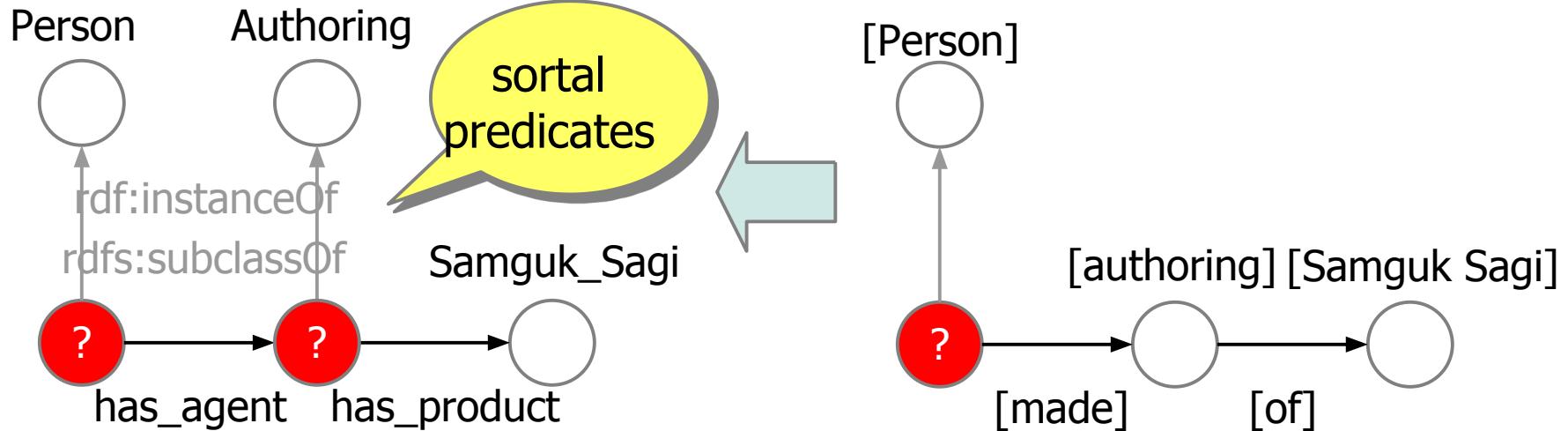
Who made the authoring of “Samguk Sagi”?



SVO 4. Instantiation



Who made the authoring of “Samguk Sagi”?





OKBQA-4

TUTORIAL : 16-17
SYMPOSIUM : 18
HACKATHON : 18-21
July 2016, Jeju, Korea



Talk 5: How to develop a controller, extension, and application cases

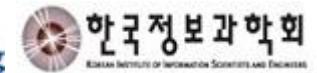
Jiseong Kim

Ph.D. Student @ KAIST

Contact: jiseong@kaist.ac.kr



UNIVERSITAT LEIPZIG



Contents

1. What is the **controller**? Is it **necessary**?
2. What is the **key functions**?
3. Controller **API & Usage**
4. **Applications with Extension**

What is “the controller”? Is it “necessary”?



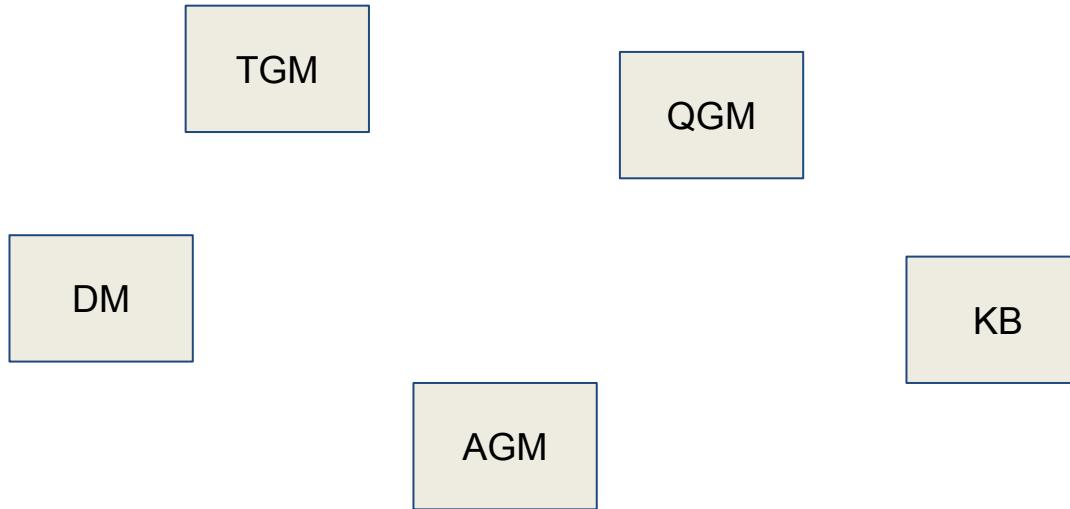
A Workflow Management System

- It provides an infrastructure for the **set-up, performance and monitoring** of a defined sequence of tasks, arranged as a workflow.
- It integrates separated tasks into a workflow with soft wiring.



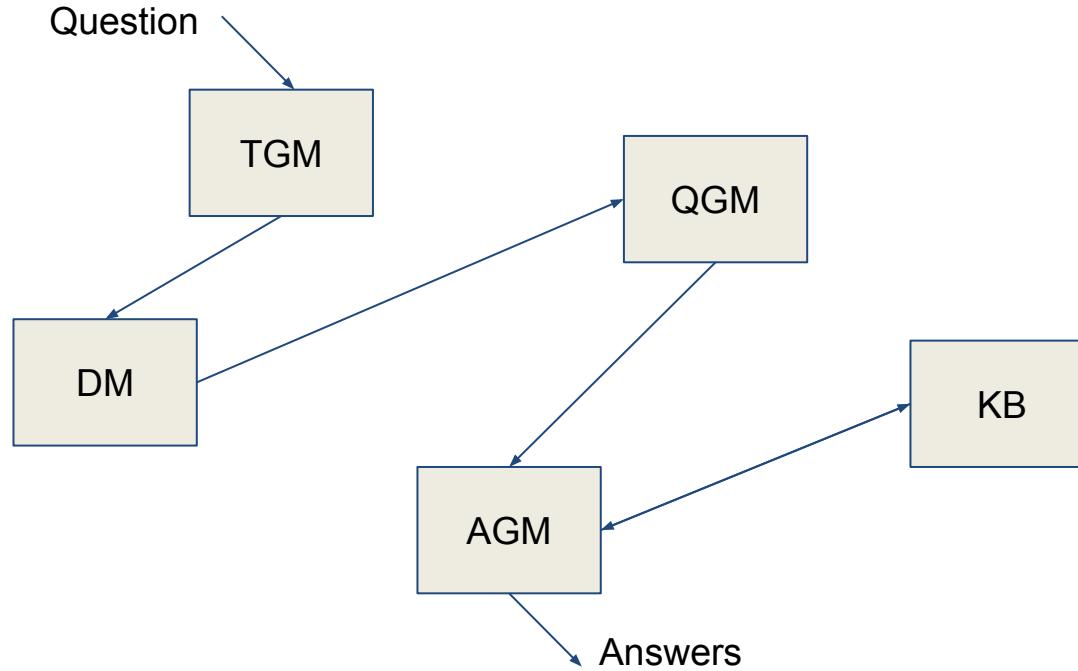
Needs of Workflow Manager for OKBQA

- Inherently, OKBQA modules are separated.
 - TGM, DM, QGM, AGM, KB, and so on



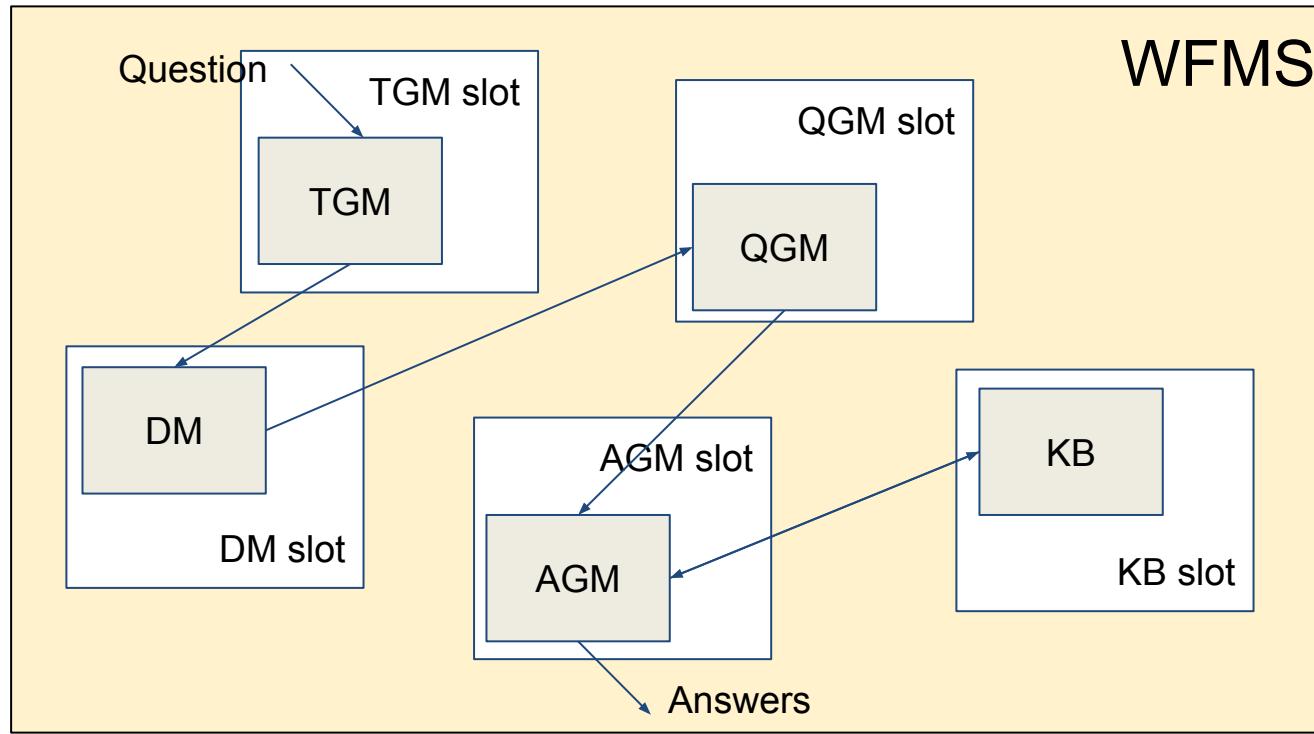
Needs of Workflow Manager for OKBQA

- They can be hard-wired.
 - Less flexibility
 - Each module hard to be replaced to better one.



Needs of Workflow Manager for OKBQA

- What if soft-wired by WFMS?

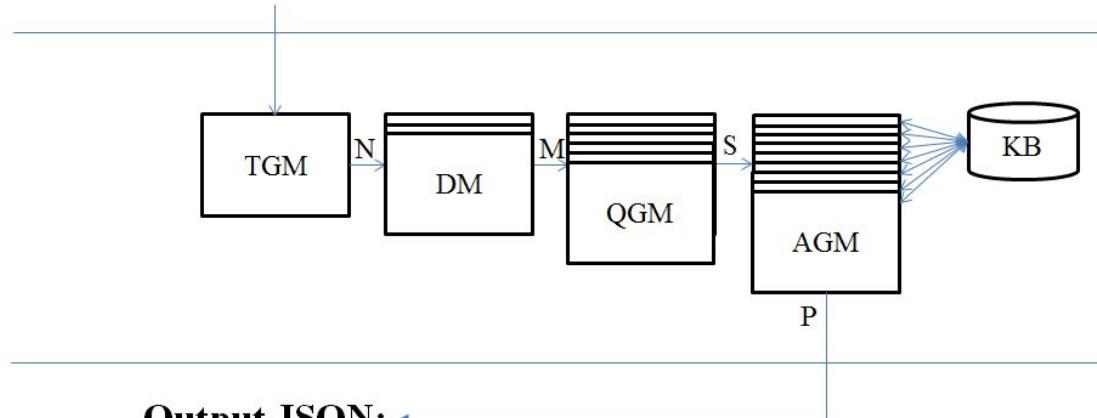


Controller: Workflow Manager for OKBQA

- A workflow management module for integrating OKBQA modules by soft-wiring
 - It interlinks I/O between TGM, DM, QGM, AGM and KB.

Input JSON:

```
{"string": "question", "language": "en"}
```



Output JSON:

```
{
  "log": [log(1), log(2), ..., log(n)],
  "answers": [answer(1), answer(2), ..., answer(n)]
}
```

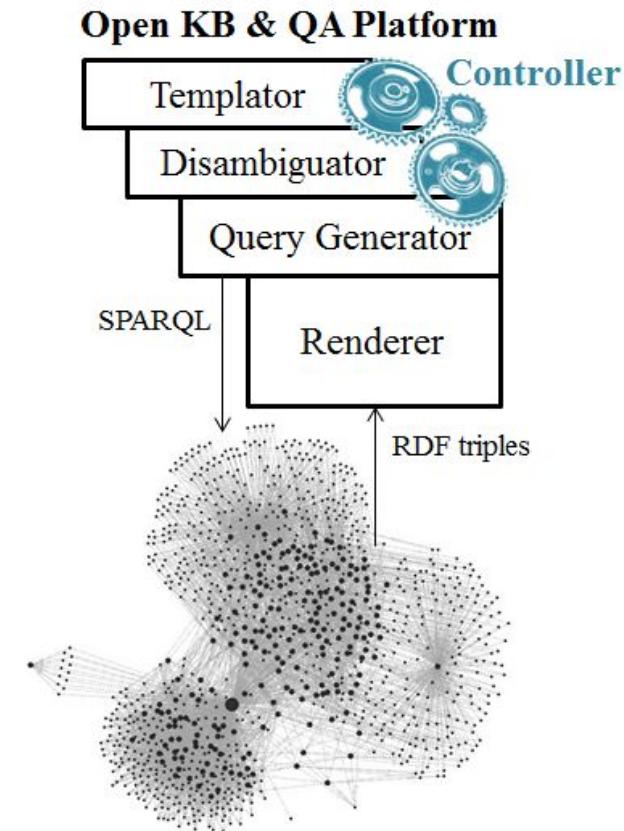
Advantages of The Dedicated WFMS

- Collaborative developments
 - **Distributed** developments
 - Each module can be developed **separately**
 - Easy **integration**
 - **I/O linking** between modules
 - After developments done,
the controller integrates modules in an united system.
 - **Logging and Alarming**
 - If integration fails, the controller **alarms** developers by **log-messages** to chase a **cause of fails**

What is the “key functions”?

Controller: Essential Functions

- The module for integrating OKBQA modules
 - It interlinks I/O between TGM, DM, QGM, AGM, and KB.
 - + RESTful API supports
 - + Fault tolerance
 - I/O formats
 - + Fault alarming (when not-tolerable)
 - Encodings
 - Exceptions
 - I/O formats (hard to be corrected)
 - + Flexible configuration
 - Module addresses
 - Graph URIs of KBs
 - Module time-out
 - + Logging
 - I/O flows
 - Answers for each SPARQL query
 - Pretty-printing



Controller API & Usage

Three level Interfaces

- API 1: Command-line
- API 2: RESTful service
- GUI: Web interface

API 1: Command-Line

- It supports command-line API to applications.
 - Usage
 - Input JSON: {“string”: x, “language”: y, “conf”: z}
 - x = “A natural language question”
 - y = “ko” / “en”
 - z = { “tgm”: [tgm_address(1), ..., tgm_address(n)],
...
“kb”: [kb_address(1), ..., kb_address(n)],
“graph_uri”: “graph_uri(1)”,
“timeout”: t }
 - Execution: python cm_terminal.py “Input JSON”
 - Output JSON: {“log”: x, “answers”: y}
 - x = [log(1), log(2), ..., log(n)]
 - y = [ans(1), ans(2), ..., ans(n)]

API 2: RESTful Service

- It provides RESTful API to the applications.
 - Built on the top of the command-line API
- Program: cm_rest.py
 - Usage
 - Input JSON: {"string": x, "language": y, "conf": z}
 - Execution: python cm_rest.py
 - Output JSON: {"log": x, "answers": y}
 - x = [log(1), log(2), ..., log(n)]
 - y = [ans(1), ans(2), ..., ans(n)]

GUI: Web Interface

- It provides GUI to the developers.
 - Built on the top of the RESTful API
 - Service: <http://ws.okbqa.org/~testuser02/>

The screenshot shows the OKBQA Web Interface. At the top, there is a blue header bar with the text "OKBQA Web Interface". Below the header, there is a search bar with the text "en" and a dropdown arrow, followed by the query "Which rivers flow through Seoul?". To the right of the search bar are icons for search, edit, export, and settings, along with a "Log" button. The main content area displays a list of URLs and a JSON representation of the input query.

http://dbpedia.org/resource/Han_River_(Korea)
http://dbpedia.org/resource/Cheonggyecheon
http://dbpedia.org/resource/Cheonggyecheon
http://dbpedia.org/resource/Ara_Canal
http://dbpedia.org/resource/Jungnangcheon
http://dbpedia.org/resource/Jungnangcheon
http://dbpedia.org/resource/Han_River_(Korea)
http://dbpedia.org/resource/Yanghwa_Bridge
http://dbpedia.org/resource/Hannam_Bridge
http://dbpedia.org/resource/Banghwa_Bridge
http://dbpedia.org/resource/Jamsu_Bridge
http://dbpedia.org/resource/Hannam_Bridge
http://dbpedia.org/resource/Banghwa_Bridge
http://dbpedia.org/resource/Jamsu_Bridge

```
[{"CM_input": {"language": "en", "string": "Which rivers flow through Seoul?"}, {"TGM_input": {"language": "en", "string": "Which rivers flow through Seoul?"}}]
```

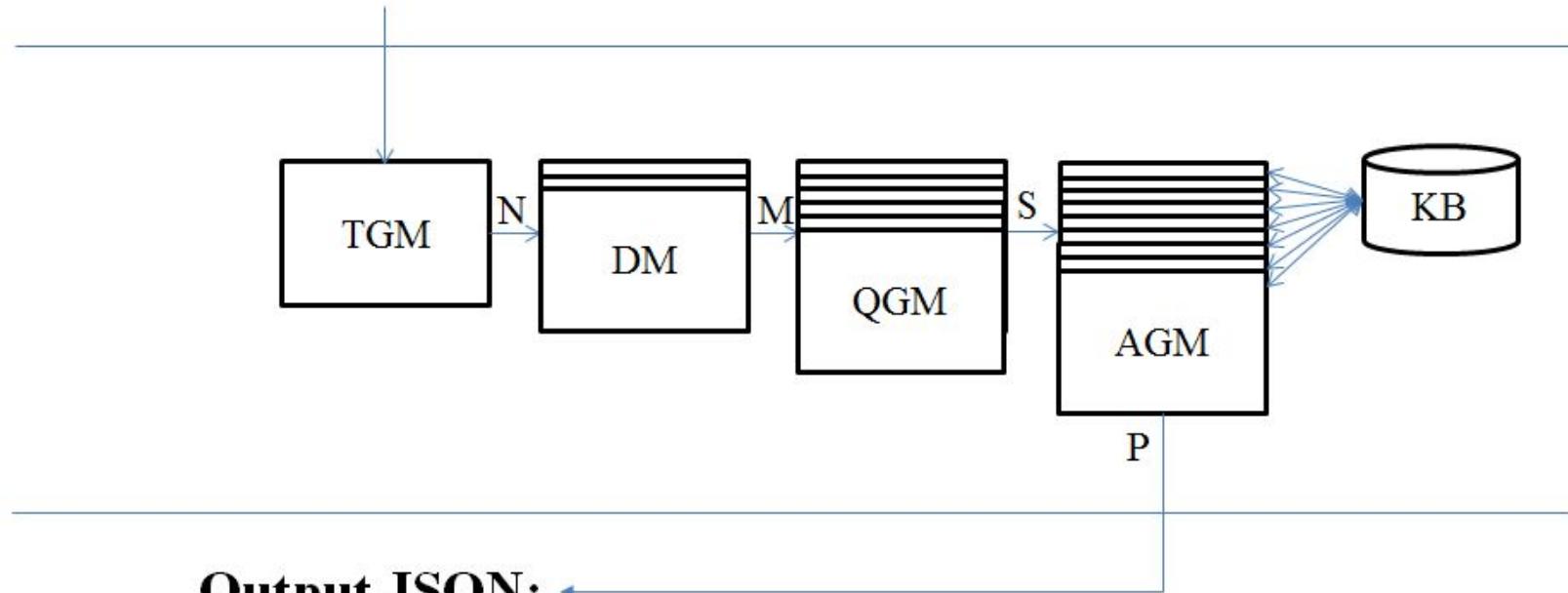


Applications with Extension

Current Architecture

Input JSON:

```
{"string": "question", "language": "en"}
```



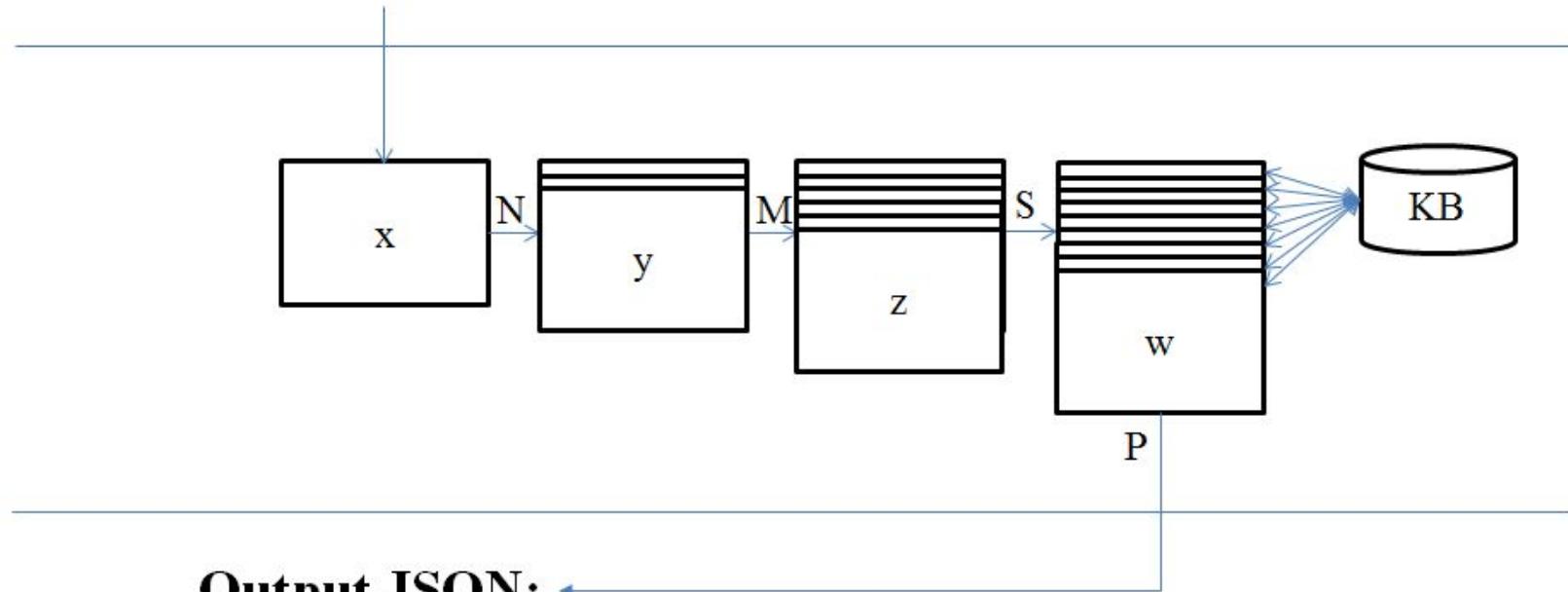
Output JSON:

```
{
  "log": [log(1), log(2), ..., log(n)],
  "answers": [answer(1), answer(2), ..., answer(n)]
}
```

Current Architecture

Input JSON:

```
{"string": "question", "language": "en"}
```



Output JSON:

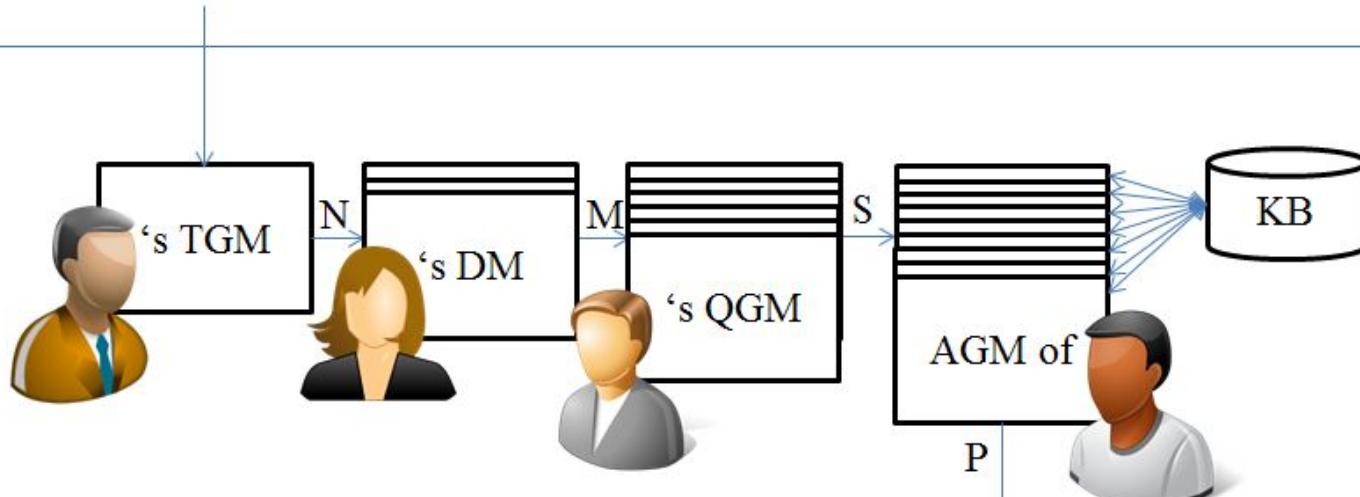
```
{
  "log": [log(1), log(2), ..., log(n)],
  "answers": [answer(1), answer(2), ..., answer(n)]
}
```



Current Architecture

Input JSON:

```
{"string": "question", "language": "en"}
```



Output JSON:

```
{
  "log": [log(1), log(2), ..., log(n)],
  "answers": [answer(1), answer(2), ..., answer(n)]
}
```

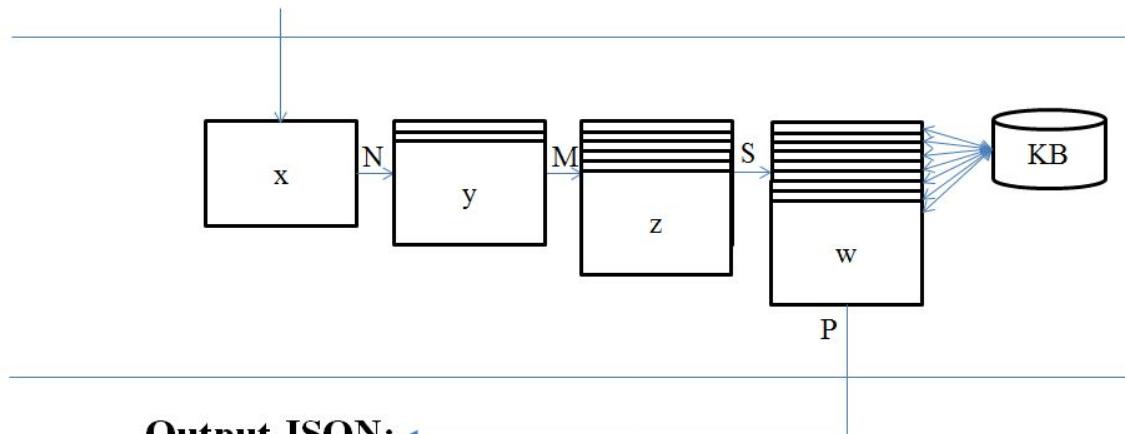


Current Architecture

- Limitations
 - Fixed module slots
 - Flowing I/O by one direction

Input JSON:

```
{"string": "question", "language": "en"}
```



Output JSON:

```
{  
    "log": [log(1), log(2), ..., log(n)],  
    "answers": [answer(1), answer(2), ..., answer(n)]  
}
```

Extension

- **Key functions** as a controller
 - I/O transfer among modules
 - Fault tolerance
 - Fault alarming
 - Flexible configuration
 - Logging for debugging
- **Possible extension** for a controller
 - Module slot # configuration
 - Specifying arbitrary # of modules
 - Link configuration
 - Specifying I/O direction between modules

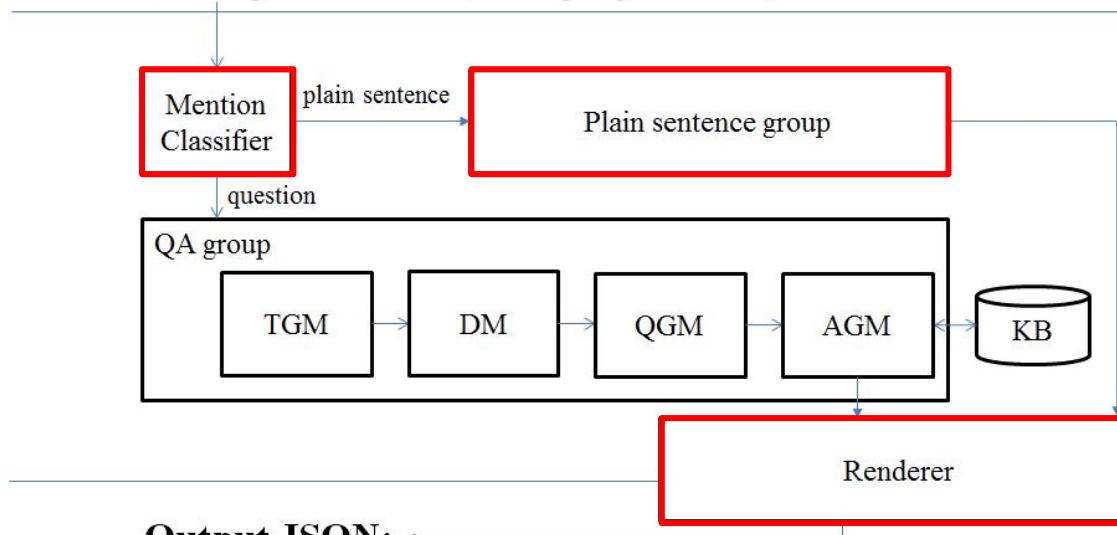


Extension 1: Module # Configuration

- For specific # of modules w.r.t. an application
 - e.g., Chatbots

Input JSON:

```
{"string": "mention", "language": "en"}
```



Output JSON:

```
{  
    "log": [log(1), log(2), ..., log(n)],  
    "string": [mention(1), mention(2), ..., mention(n)]  
}
```

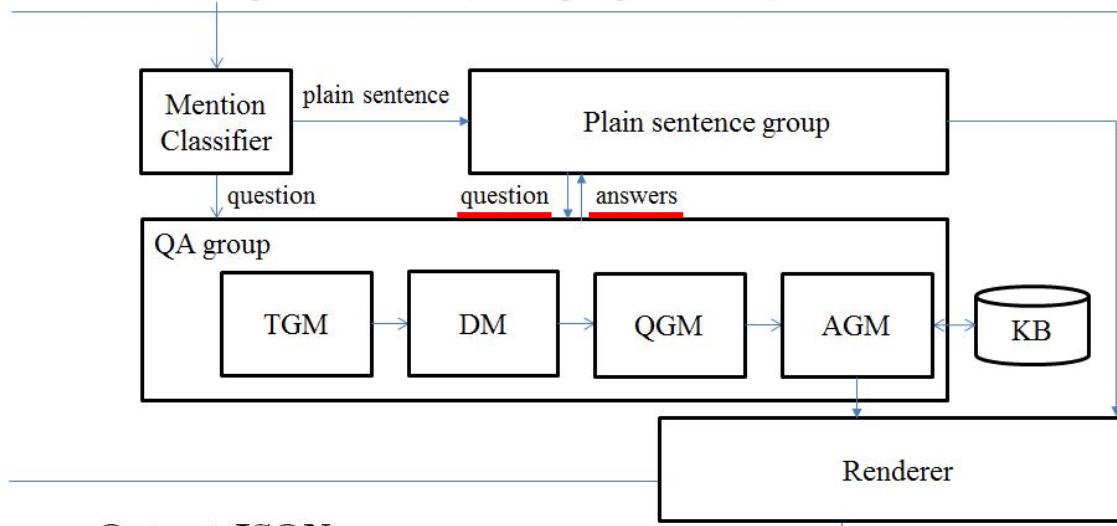


Extension 2: Link Configuration

- All applications are not like waterfall.
 - e.g., Chatbots with *self-questioning*

Input JSON:

```
{“string”: “mention”, “language”: “en”}
```



Output JSON:

```
{
  “log”: [log(1), log(2), ..., log(n)],
  “string”: [mention(1), question(2), ..., answer(n)]
}
```



Remind: Essential Attributes

- Collaborative developments
 - **Distributed** developments
 - Each module can be developed **separately**
 - Easy **integration**
 - After developments done,
the controller integrates modules
 - **I/O linking** between modules
 - **Logging** and **Alarming**
 - If integration fails, the controller **alarms** developers
by **log-messages** to chase a **cause of fails**



Available Modules

- English
 - TGM
 - <http://ws.okbqa.org:1555/templategeneration/templator/>
 - DM
 - <http://ws.okbqa.org:2357/agdistis/run>
 - QGM
 - <http://ws.okbqa.org:38401/queries>
 - AGM
 - <http://ws.okbqa.org:7744/agm>
 - KB
 - <http://dbpedia.org/sparql>

4.okbqa.org



<http://coling2016.okbqa.org>

- OKBQA workshop in COLING 2016
- Dec. 11 (Sun)
- Osaka

