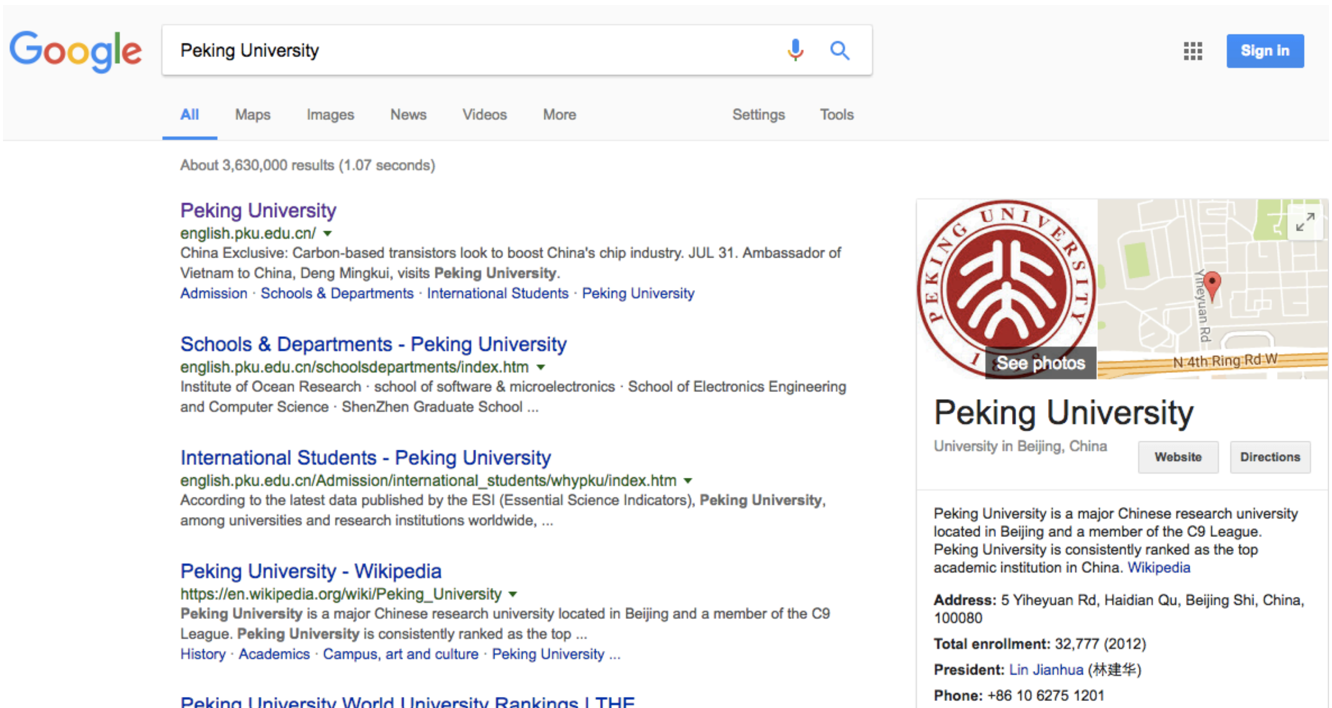
# Question Answering Over Knowledge Graph

Lei Zou

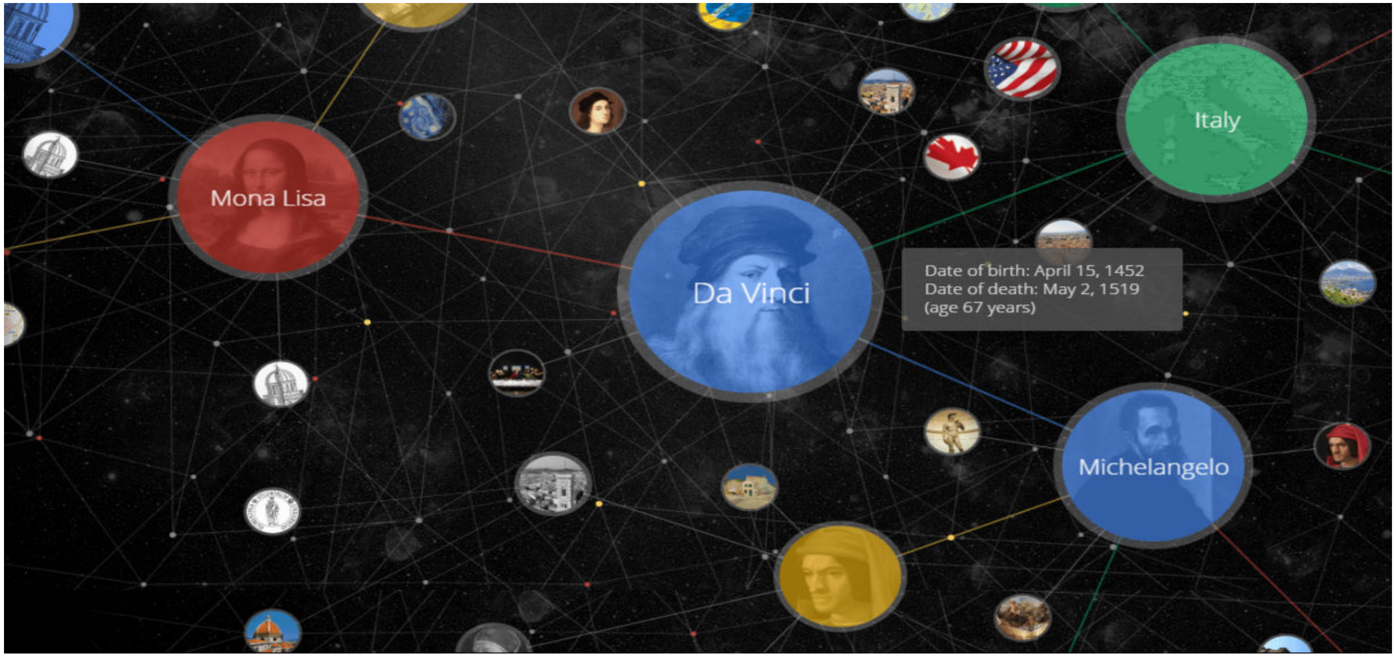
# Knowledge Graph

Google launches **Knowledge Graph** project at 2012.



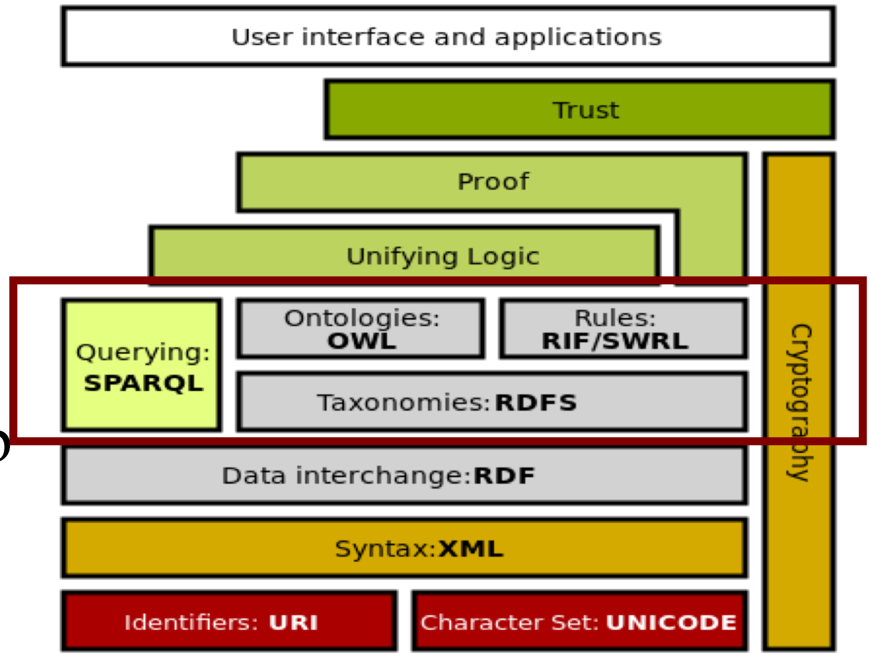
# Knowledge Graph

Essentially, KG is a sematic network, which models **the entities (including properties) and the relation between each other.**



# Resource Description Framework (RDF)

- RDF is an **de facto standard** for Knowledge Graph (KG).
- RDF is a **language** for the conceptual modeling of information about web resources
- A **building block** of semantic web
- Make the information on the web and the interrelationships among them "**Machine Understandable**"



# RDF & SPARQL

## RDF Datasets

| Subject | Predicate | Object |
|---|---|---|
| Resident_Evil:_Retribution | type | film |
| Resident_Evil:_Retribution | budget | “6.5E7" |
| Resident_Evil:_Retribution | director | Paul_W._S._Anderson |
| Paul_W._S._Anderson | type | director |
| Resident_Evil | director | Paul_W._S._Anderson |
| Paul_Anderson_(actor) | type | actor |
| The_Revenant | strarring | Philadelphia |
| Priestley Medal | awards | Paul S. Anderson |
| Maclovia_(1948_film) | distributor | Filmex |

“What is the budget of the film directed by Paul Anderson ?.”

**SPARQL**

```
SELECT ?y WHERE
{
?x director Paul_W._S._Anderson .
?x type film .
?x budget ?y.
}
```

# **A Fundamental Problem**：How to store RDF data and answer SPARQL queries

| Subject | Predicate | Object |
|---|---|---|
| Abraham_Lincoln | hasName | “Abraham Lincoln" |
| Abraham_Lincoln | BornOnDate | “1809-02-12" |
| Abraham_Lincoln | DiedOnDate | “1865-04-15"” |
| Abraham_Lincoln | DiedIn | Washington_DC |
| Abraham_Lincoln | bornIn | Hodgenville KY |
| Reese_Witherspoon | bornOnDate | "1976-03-22" |
| Reese_Witherspoon | bornIn | New_Orleans_LA |
| New_Orleans_LA | foundingYear | “1718” |
| New Orleans LA | locatedIn | United_States |
| United_States | hasName | “United States” |
| United_States | hasCapital | Washington_DC |
| United_States | foundingYear | “1776” |

## SPARQL

```
SELECT ?name
WHERE {
?m <bornIn> ? c i t y .
?m <hasName> ?name .
?m <bornOnDate> ?bd .
? c i t y <foundingYear> ` `1718 ' ' .
FILTER( regex (str (?bd ), “1 9 7 6 ' ' ) )
}
```

**How to answer SPARQL efficiently.**

# Existing Solutions: Resorting to **RDBMS** techniques

| Subject | Predicate | Objects |
|---|---|---|
| Abraham_Lincoln | hasName | “Abraham Lincoln" |
| Abraham_Lincoln | BornOnDate | “1809-02-12" |
| Abraham_Lincoln | DiedOnDate | “1865-04-15"” |
| Abraham_Lincoln | DiedIn | Washington_DC |
| Abraham_Lincoln | bornIn | Hodgenville KY |
| Reese_Witherspoon | bornOnDate | "1976-03-22" |
| Reese_Witherspoon | bornIn | New_Orleans_LA |
| New_Orleans_LA | foundingYear | “1718” |
| New Orleans LA | locatedIn | United_States |
| United_States | hasName | “United States” |
| United_States | hasCapital | Washington_DC |
| United_States | foundingYear | “1776” |

**SPARQL**

SELECT ?name
WHERE {
?m <bornIn> ? c i t y .
?m <hasName> ?name .
?m <bornOnDate> ?bd .
? c i t y <foundingYear> ` `1718 ' ' .
FILTER( regex (str (?bd ), “1 9 7 6 ' ' ) )
}

**SQL**

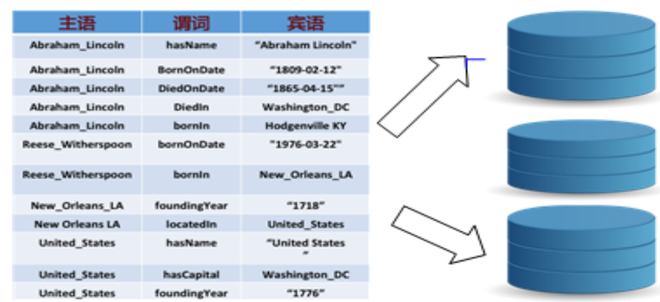
SELECT T2 . o b j e c t
F
T
WHERE T1.pro bornIn "
AND T2.proper hasName"
AND T3.propert = "bornOnDate "
AND T1.subject=T2.subject
AND T2.subject=T3.subject
AND T1.object=T4.subject
AND T4.propety=“foundingYear “
AND T4.object=" 1718 "
AND T3.object LIKE '%1976%'

**Too many self-joins**

# Existing Solutions (based on relational techniques)

- **Property Table** Jena [Wilkinson et al., 2003] ,FlexTable [Wang et al., 2010] , DB2-RDF [Bornea et al., 2013]
- **Vertically partitioned tables** SW-store [Abadi et al., 2009]
- **Exhaustive indexing** RDF-3X [Neumann and Weikum, 2008], Hexastore [Weiss et al., 2008]

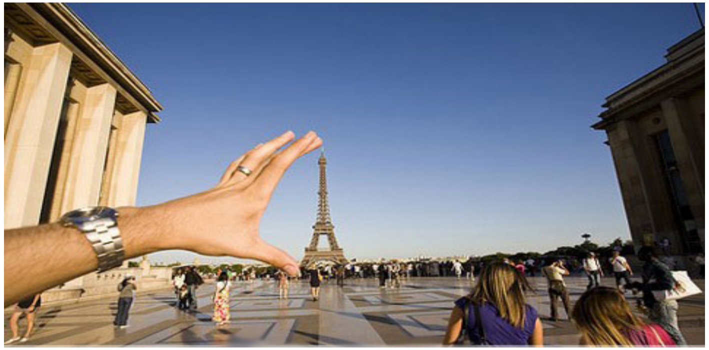
Basic Ideas: dividing the large single triple-table into several carefully-designed tables.

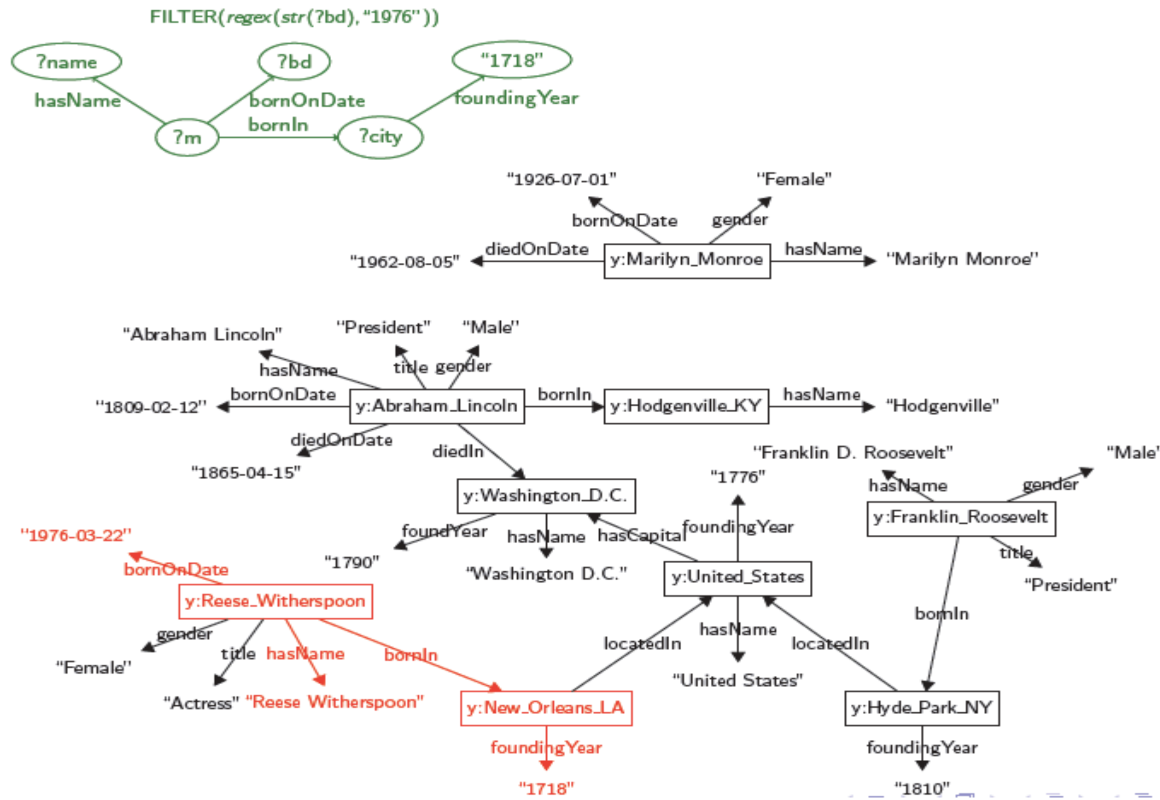


| 主语 | 谓词 | 宾语 |
| --- | --- | --- |
| Abraham_Lincoln | hasName | "Abraham Lincoln" |
| Abraham_Lincoln | BornOnDate | "1809-02-12" |
| Abraham_Lincoln | DiedOnDate | "1865-04-15" |
| Abraham_Lincoln | DiedIn | Washington_DC |
| Abraham_Lincoln | bornIn | Hodgenville KY |
| Reese_Witherspoon | bornOnDate | "1976-03-22" |
| Reese_Witherspoon | bornIn | New_Orleans_LA |
| New_Orleans_LA | foundingYear | "1718" |
| New Orleans LA | locatedIn | United_States |
| United_States | hasName | "United States" |
| United_States | hasCapital | Washington_DC |
| United_States | foundingYear | "1776" |

- M. T. Özsu. "A Survey of RDF Data Management Systems", Front. Comp. Sci., 2016.
- Lei Zou, M. T. Özsu. “Graph-based RDF Data Management”, Data Science and Engineering, 2(1): 56-70 (2017)

# Our Solution---gStore [Zou et al., VLDB 11; VLDB J 14 ]






Answering SPARQL == subgraph matching

# Our Research Roadmap

gAnswer: Answering Natural Language Question by graph matching

gStore-D: Distributed RDF Graph Database

gStore: Graph Matching-based SPARQL Engine

Our Focus

**Subgraph Search**

Structure - aware Graph Data Indexing and Query Optimization

Modeling

| 主语 | 谓词 | 宾语 |
|---|---|---|
| Abraham_Lincoln | hasName | “Abraham Lincoln" |
| Abraham_Lincoln | BornOnDate | “1809-02-12" |
| Abraham_Lincoln | DiedOnDate | “1865-04-15"” |
| Abraham_Lincoln | DiedIn | Washington_DC |
| Abraham_Lincoln | bornIn | Hodgenville KY |
| Reese_Witherspoon | bornOnDate | "1976-03-22" |
| Reese_Witherspoon | bornIn | New_Orleans_LA |
| New_Orleans_LA | foundingYear | “1718” |
| New Orleans LA | locatedIn | United_States |
| United_States | hasName | “United States” |
| United_States | hasCapital | Washington_DC |
| United_States | foundingYear | “1776” |

RDF Datasets

SPARQL

```
SELECT ?name
WHERE {
?m <bornIn> ? c i t y .
?m <hasName> ?name .
?m <bornOnDate> ?bd .
? c i t y <foundingYear> ``1718 '' .
FILTER( regex (str (?bd ), “1 9 7 6 '' ) )
}
```

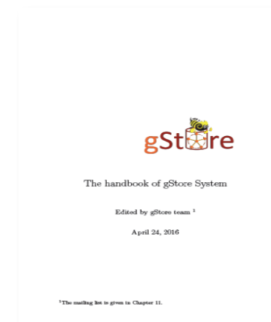
自然语言问题

“杨绛爱人的著作有哪些？”
“Who was married to an actor that play in Philadelphia ? ”

# Our System

**Codes**：More than 140,000 lines C++, coding from scratch

**Project Address:**
[https://github.com/pkumod/gStore](https://github.com/pkumod/gStore)
including all codes; user manual; benchmarking test report; system demo video.


The handbook of gStore System

Edited by gStore team [1]

April 24, 2016

[1] The mailing list is given in Chapter 11.

**Licenses**: BSD
**Deployment**: C/S or single machine
**API**: C++, Java, Phython, PHP and HTTP Rest

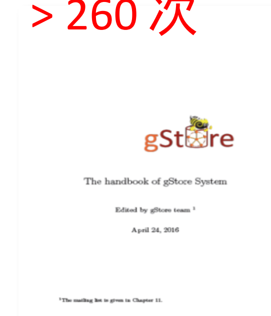
supporting SPARQL 1.1 (including UNION, OPTIONAL, FILTER, GROUP BY, BIND)

# Our System gStore

**Codes**：More than 140,000 lines C++, coding from scratch

**Project Address:**

GitHub月均下载 ＞260 次


The handbook of gStore System

Edited by gStore team

April 24, 2016

[https://github.com/pkumod/gStore](https://github.com/pkumod/gStore)

including all codes; user manual; benchmarking test report; system demo video.

**Licenses**: BSD

**Deployment**: C/S or single machine

**API**: C++, Java, Phython, PHP and HTTP Rest

supporting SPARQL 1.1 (including UNION, OPTIONAL, FILTER, GROUP BY, BIND)

# SPARQL Endpoint

http://freebase.gstore-pku.com/

gStore

About API

Database Name

freebase

Query Examples

q1

Query Text

```
select ?s where
{
    ?s <http://www.w3.org/2000/01/rdf-schema#label>   "footballdb ID"@en .
    ?s <http://rdf.freebase.com/ns/type.property.schema>     <http://rdf.freebase.com/ns/american_football.football_
    ?s <http://rdf.freebase.com/ns/type.property.unique> "true" .
}
```

gStore

了解更多关于API的用法

● **Java API使用示例**

```java
import jgsc.GstoreConnector;

public class JavaAPIExample {
    public static void main(String[] args)
    {
        //初始化一个GstoreConnector类，IP地址可以是域名形式
        GstoreConnector gc = new GstoreConnector("dbpedia.gstore-pku.com", 80);
         //赋值查询的SPARQL语句
         String sparql = "SELECT ?v0 WHERE "
                  + "{"
                  +"?v0 <http://dbpedia.org/ontology/associatedBand> <http://dbpedia.org/resource/LCD_Soundsystem> ."
                  +"}";
        //调用gc的query(_username, _password, _db_name, _sparql)方法得到查询结果
         String answer = gc.query("endpoint", "123", "dbpedia_full", sparql);
        System.out.println(answer);
    }
}
```

**注：**

**1. 我们提供的查询终端服务均开放了用户：_username=endpoint, _password=123，可通过此用户名和密码调用gc.query()函数**

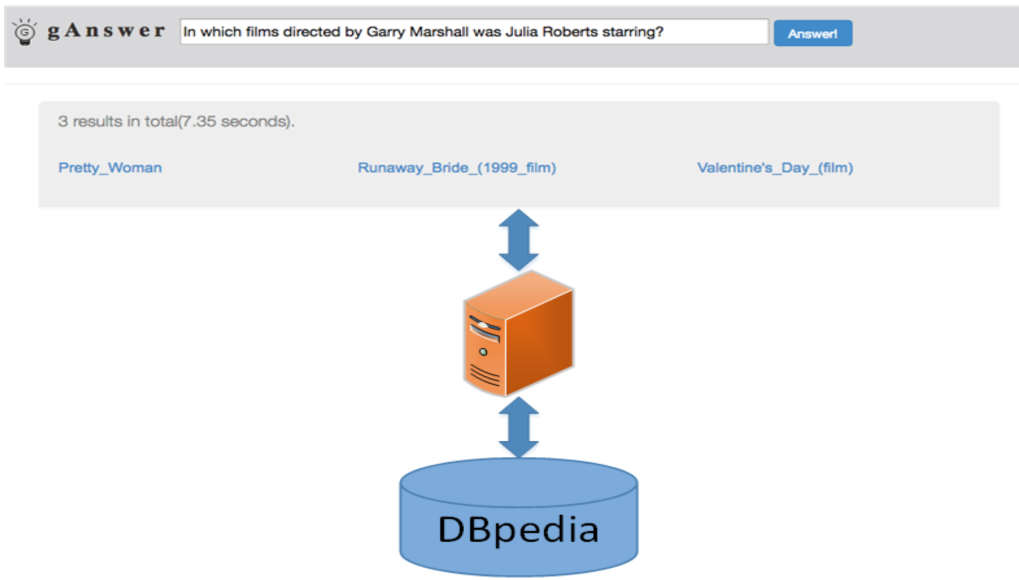
**2. 我们只对用户开放查询权限，不开放对数据库的更新权限，INSERT, DELETE操作将返回失败**

**3. 为保护数据的所有权，避免恶意下载数据，我们限制查询结果数为十万，超过数量限制的查询将只返回前十万条结果**

月均调用 >6万次

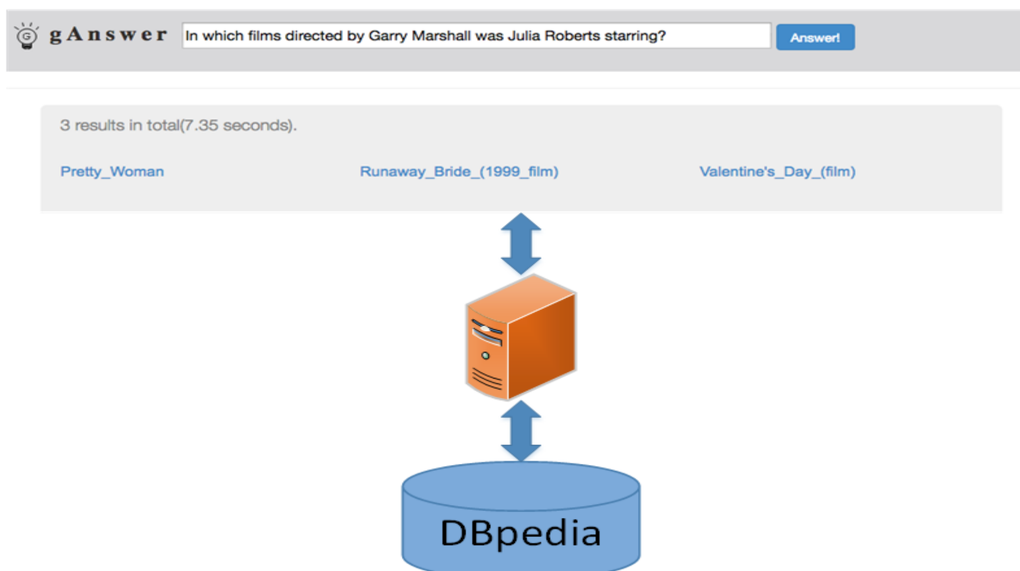
# KG-based Question/Answering

- SPARQL syntax are too complex for ordinary users
- RDF KG is"schema-less"data, not like schema-first relational database.



# KG-based Question/Answering

- An **Easy-to-Use** Interface to Access Knowledge Graph
- It is interesting to both **academia** and **industry**.
- **Interdisciplinary research** between database and NLP (natural language processing) communities.



# KG-based Question/Answering


COMMENT

Growth in genome screening could cause dangerous meddling p.27 | How genes and culture have shaped our ability to cooperate p.29 | Debating how life got going on the early Earth p.30 | Wildlife paintings from Yukon to Yellowstone p.32



## Search needs a shake-up

On the twentieth anniversary of the World Wide Web's public release, **Oren Etzioni** calls on researchers to think outside the keyword box and improve Internet trawling.

Two decades after Internet pioneer Tim Berners Lee introduced his World Wide Web project to the world using the alt.hypertext newsgroup, web search is on the cusp of a profound change — from simple document retrieval to question answering. Instead of poring over long lists of documents that contain requested keywords, users need direct answers to their questions. With sufficient scientific and financial investment, we could soon view today's keyword searching with the same nostalgia and amusement reserved for bygone technologies such as electric typewriters and vinyl records.

But this transformation could be unreasonably delayed. As a community, computer scientists have underinvested in tools that can synthesize sophisticated answers to questions, and have instead focused on incremental progress in lowest-common-denominator search. The classic keyword search box exerts a powerful gravitational pull. Academics and industry researchers need to achieve the intellectual 'escape velocity' necessary to revolutionize search. They must invest much more in bold strategies that can achieve natural-language searching and answering, rather than providing the electronic equivalent of the index at the back of a reference book.

Today, that 'book' is distributed over billions of web pages of uneven quality, and much effort has been directed at ranking the most useful results. Such engines readily index billions of documents, but overwhelm their users with millions of results in response to simple queries. This quandary only worsens as the number of web pages ▸





Oren Etzioni, AAAI Fellow

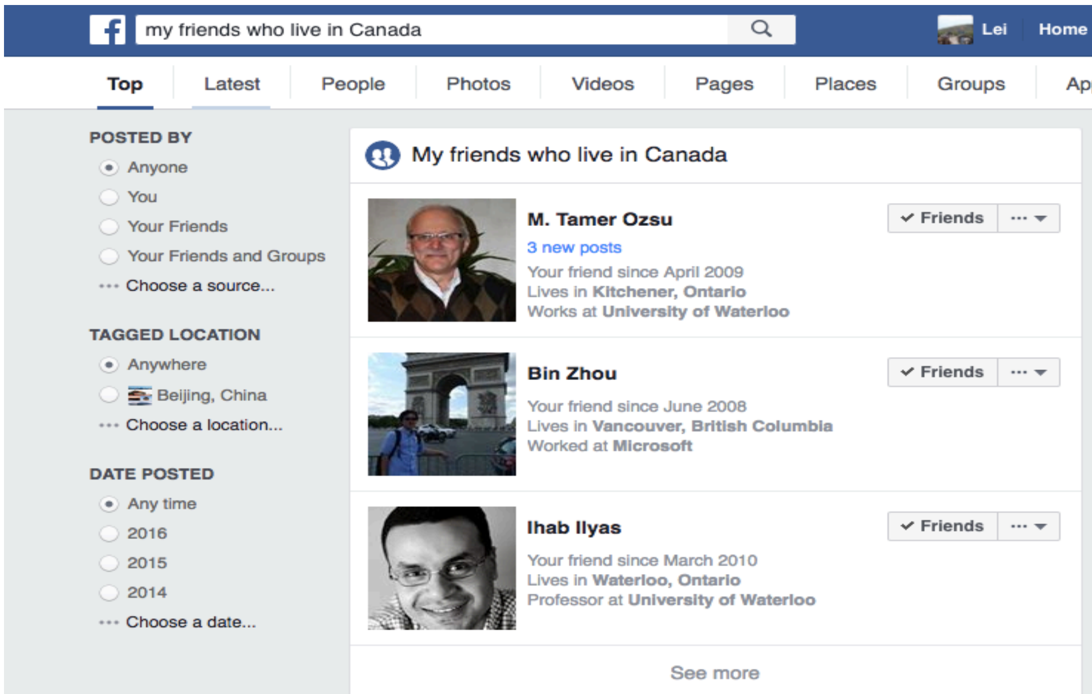
"(Researchers) They must invest much more in bold strategies that can achieve **natural-language searching and answering**"

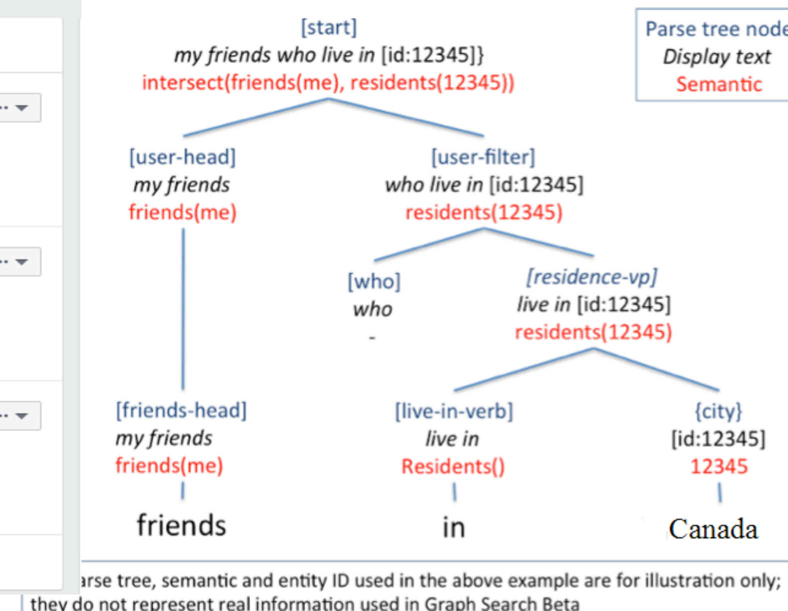
---Oren Etzioni, Search needs a shake up, **NATURE**, Vol 476, p25-26, 2011.

# Facebook Graph Search

## "My friends who live in Canada"

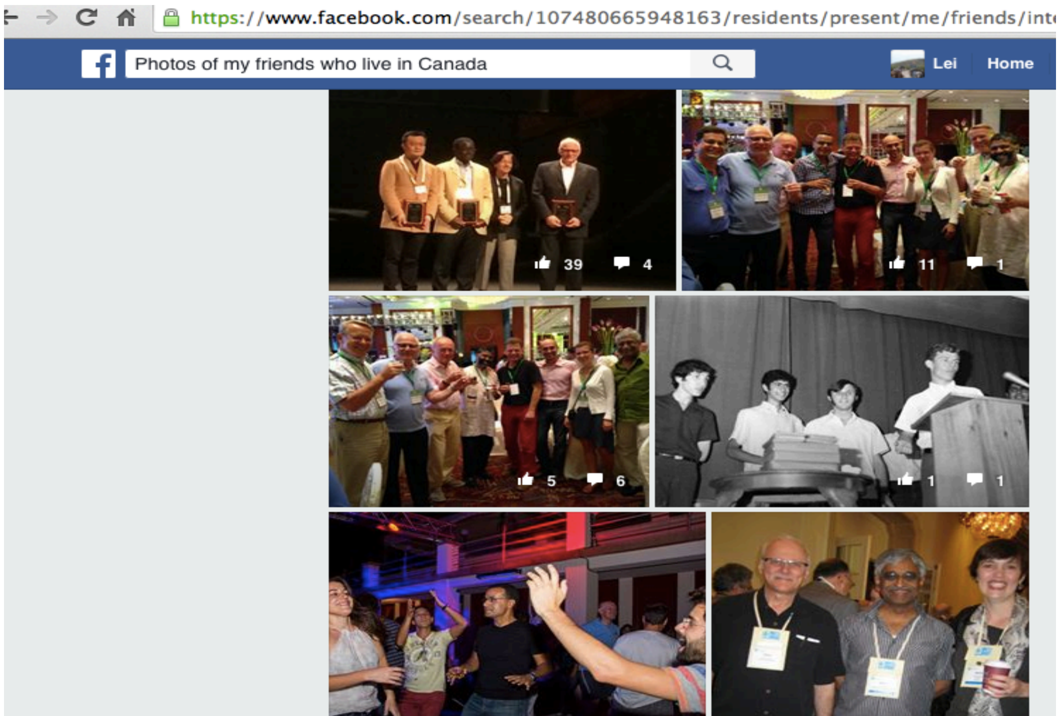
"Facebook Graph Search" ------announced by Mark Zuckerberg on January 16, 2013



# Facebook Graph Search

"Photos of my friends who live in Canada"



# EVI---(originally, True Knowledge)


| | Venture Capital |
|---|---|
| 2007-09 | 1.2 Million USD |
| 2008-07 | 4 Million USD |
| 2012-01 | Acquired by Amazon |

William Tunstall-Pedoe: *True Knowledge: Open-Domain Question Answering using Structured Knowledge and Inference*. AI Magazine 31(3): 80-92 (2010)

# Background

- Question Answering
  - Knowledge-based QA (or KBQA)
  - Document-based QA (or DBQA)
  - Community QA (or CQA)

- Types of questions
  - Factoid: *"Who is the wife of Donald Trump?"*
  - Definition: *"What is deep learning?"*
  - Yes-No: *"Is Detective Conan still serialized?"*
  - Opinion: *"What do most Chinese think of Frog Worship?"*
  - Comparison: *"What are the difference between MI and iPhone?"*
  - …

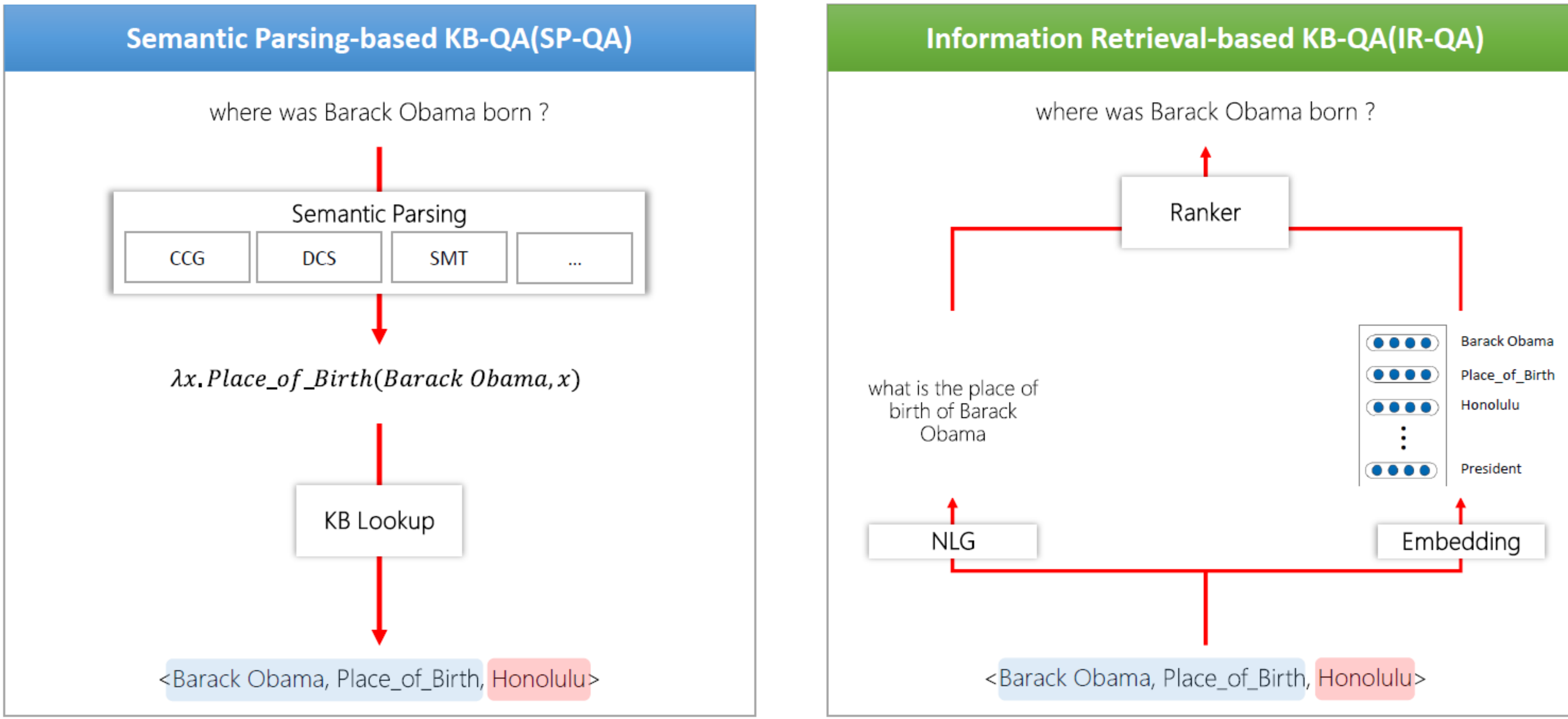
# KG-based Question/Answering

- Information Retrieval-based
  - Generate candidate answers
  - Ranking

- Semantic Parsing-based
  - Translate NLQ to logical forms
  - Executing

# Knowledge-based QA (KB-QA)

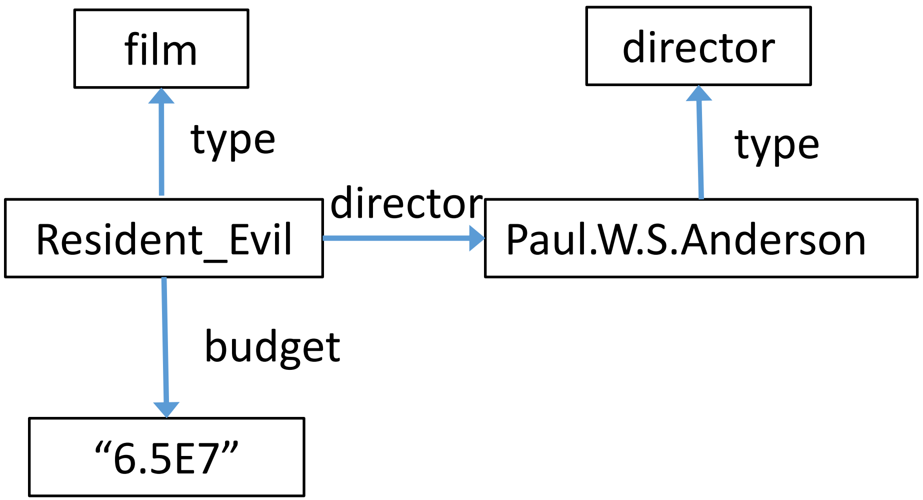
**CCG**: Combinatory Categorial Grammar
**DCS**: Dependency-based Compositional Semantics
**SMT**: Statistical Machine Translation




(Cite: Nan Duan, MSRA)

# KG-based Question/Answering

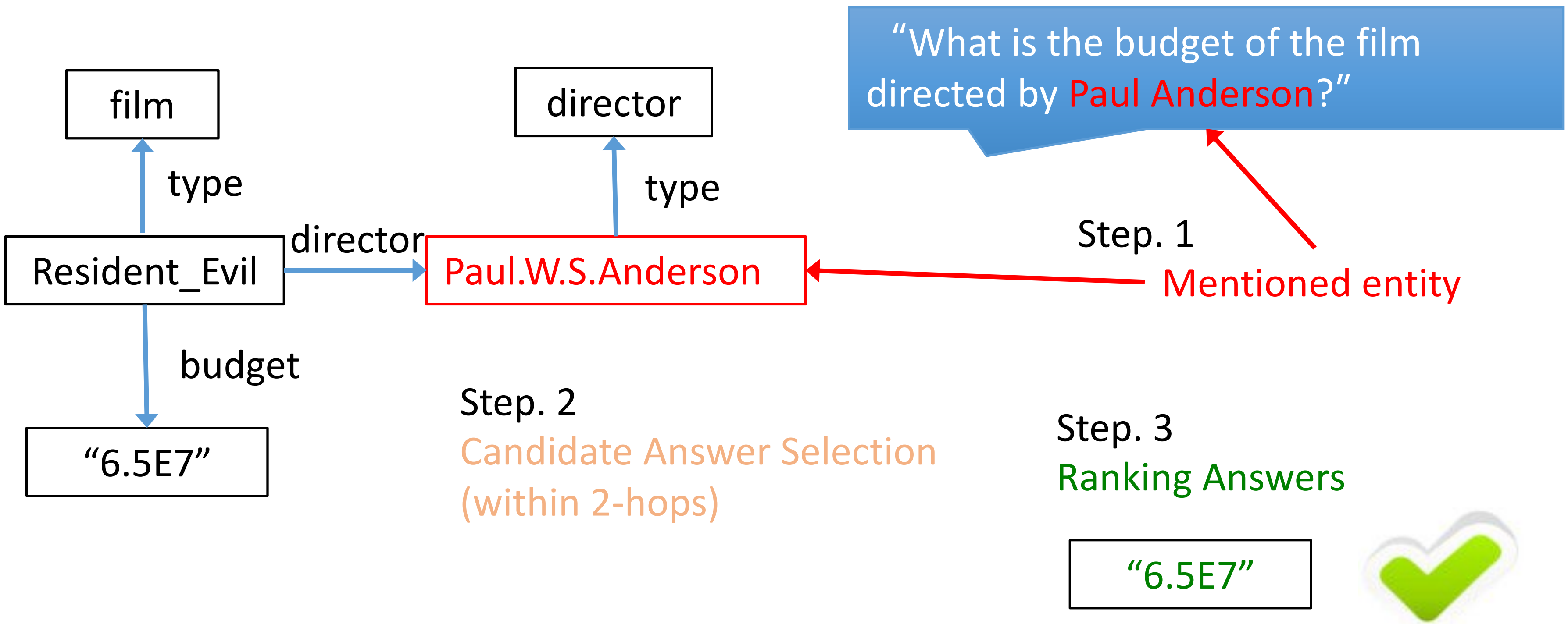
- Information Retrieval-based




"What is the budget of the film directed by Paul Anderson?"

# KG-based Question/Answering

- Information Retrieval-based



# Question Answering with Subgraph Embeddings [Bordes et al. EMNLP 2014]

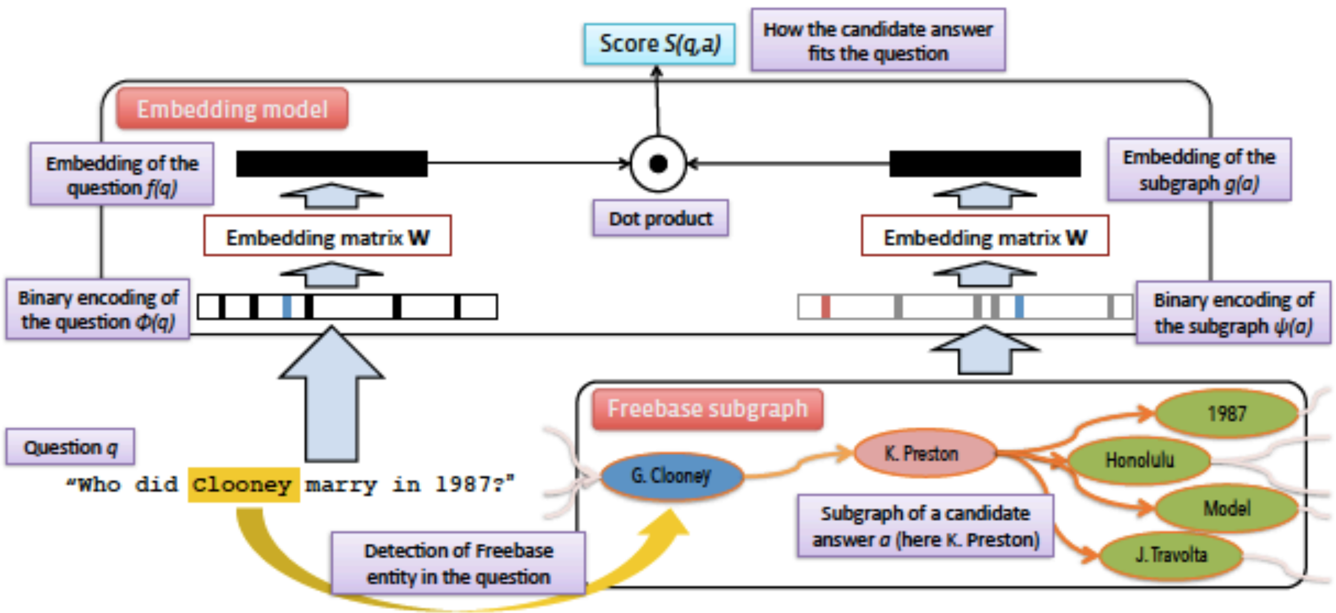



Figure 1: Illustration of the subgraph embedding model scoring a candidate answer: (i) locate entity in the question; (ii) compute path from entity to answer; (iii) represent answer as path plus all connected entities to the answer (the subgraph); (iv) embed both the question and the answer subgraph separately using the learnt embedding vectors, and score the match via their dot product.

# Question Answering with Subgraph Embeddings [Bordes et al. EMNLP 2014]

Let $W$ be a matrix $\Re^{k \times N}$

k: the dimension of the embedding space

N: $N = N_W + N_S$

$N_W$ is the number of words

$N_S$ is the number of entities and relation types

Embedding a question q

$$f(q) = W\phi(q)$$

$\phi(q)$ is a sparse vector indicating the presence of words (usually 0 or 1).

# Question Answering with Subgraph Embeddings [Bordes et al. EMNLP 2014]

Embedding a candidate answer a

$$g(a) = W\varphi(a)$$

$\varphi(a)$ is a sparse vector representation of the answer *a*

- Single Entity

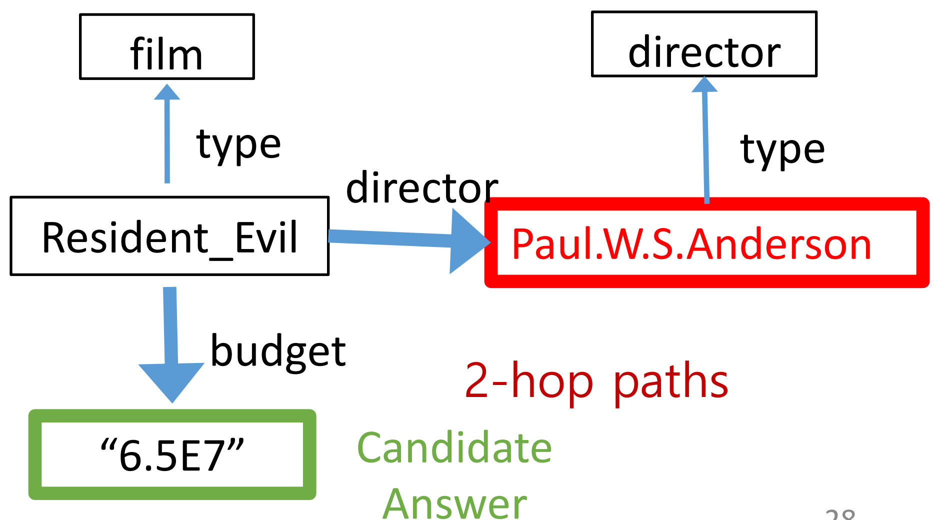
The answer is represented as a single entity:
$\varphi(a)$ is a 1-of-Ns coded vector with 1 corresponding the answer.

N= N1+N2
N1: The number of nodes in RDF graph;
N2 ： The number of distinct predciates

- Path Representation

The answer is represented as a path from the entity mentioned in the question to the answer entity *a*.

$\varphi(a)$ is a 3-of-Ns (or 4-of-Ns) coded vector, expressing the start and the end entities of the path and the relation types (but not entities) in-between.



# Question Answering with Subgraph Embeddings [Bordes et al. EMNLP 2014]

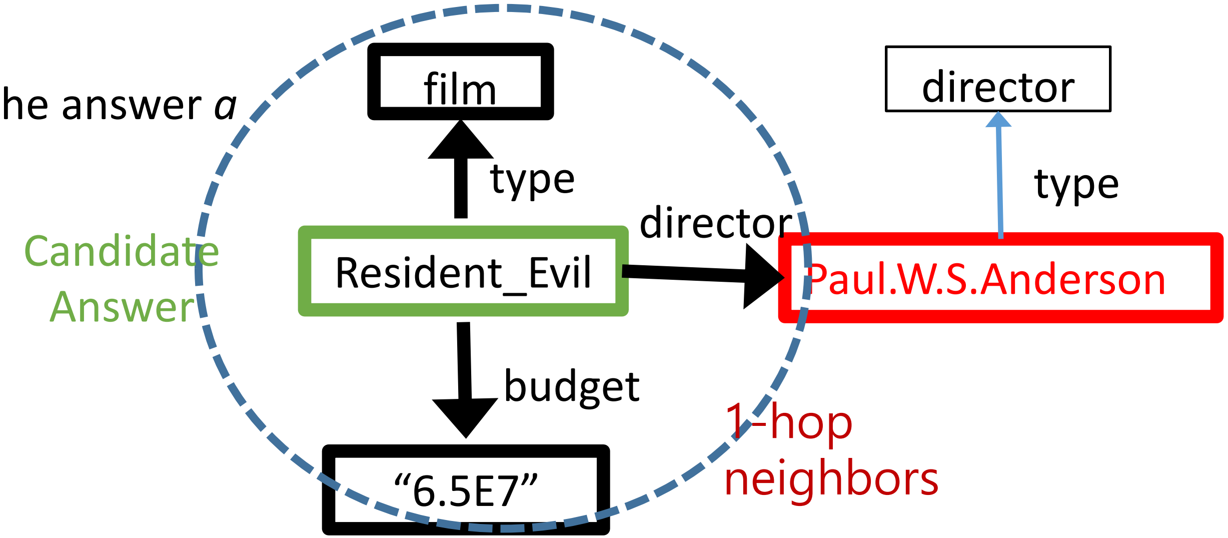
- Subgraph Representation

The answer is represented both the path and 1-hop neighbors around the answer a.

Embedding a candidate answer a

$$g(a) = W\varphi(a)$$

$\varphi(a)$ is a sparse vector representation of the answer *a*



# Question Answering with Subgraph Embeddings [Bordes et al. EMNLP 2014]

Scoring Function

candidate answer

$$S(q,a) = f(q)^T g(a)$$

question sentence

The loss function

$$\sum_{i=1}^{|D|} \sum_{a' \in A'(a_i)} \max\{0, m - S(q_i, a_i) + S(q_i, a')\}$$

$A'(a_i)$ is a set of incorrect canidates to question $q$.

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

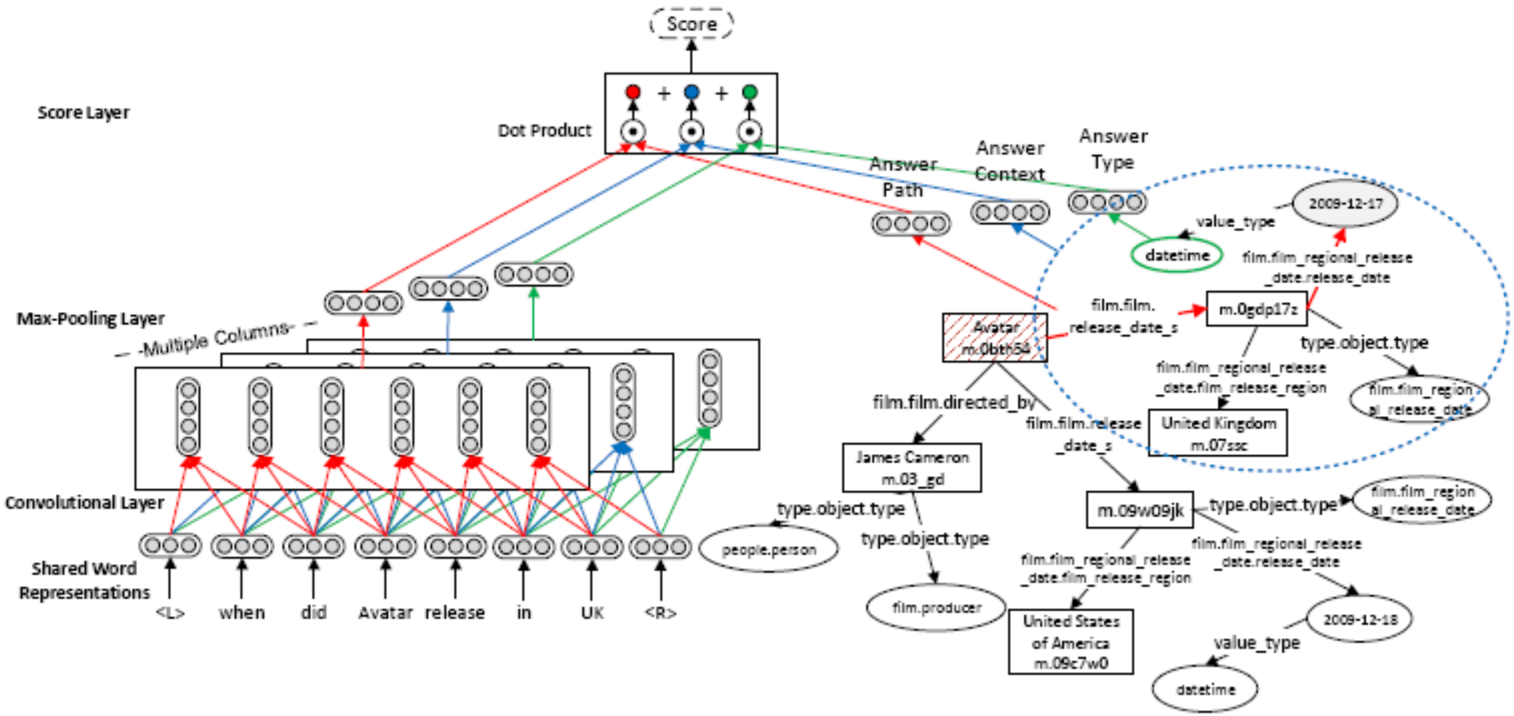



Figure 1: Overview for the question-answer pair *(when did Avatar release in UK, 2009-12-17)*. Left: network architecture for question understanding. Right: embedding candidate answers.

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

## Scoring Function

question sentence

candidate answer

$$S(q,a) =$$

$$\underbrace{f_1(q)^T g_1(a)}_{\text{answer path}} + \underbrace{f_2(q)^T g_2(a)}_{\text{answer context}} + \underbrace{f_3(q)^T g_3(a)}_{\text{answer type}}$$

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

MCCNNs for
Question Understanding

Let the question $q = w_1 w_2 \ldots w_n$

The look layer transform every word into a vector

$$w_j = W_v u(w_j)$$

$W_v \in \Re^{d_v \times |V|}$,

$d_v$ is the word embedding dimention and

|V| is the vocabulary size

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

MCCNNs for
Question Understanding

Let the question $q = w_1 w_2 \ldots w_n$

The convolutional layer computes representation of the words in sliding windows.

$$x_j = h(W[w_{j-s}^T \ldots w_j^T \ldots w_{j+s}^T] + b)$$

The max-pooling layer

$$f(q) = \max_{j=1,\ldots,n} \{x_j\}$$

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

## Embedding Candidate Answers

### Answer Path

$$g_1(a) = \frac{1}{\left\| u_p(a) \right\|_1} W_p u_p(a)$$

$u_p(a)$ is a length-|R| binary vector, indicating the presence or absence of every relation in the answer path.

$W_p \in \Re^{d_q \times |R|}$ is the parameter matrix

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

## Embedding Candidate Answers

### Answer Context

The 1-hop entities and relations connected to the answer path are regarded as the *answer context*.

$$g_2(a) = \frac{1}{\|u_c(a)\|_1} W_c u_c(a)$$

$u_c(a)$ is a length-|C| binary vector, indicating the presence or absence of every entity or relation in the context.

$W_c \in \Re^{d_q \times |C|}$ is the parameter matrix

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

## Embedding Candidate Answers

### Answer Type

Type information is an important clue to score candidate answers.

$$g_3(a) = \frac{1}{\|u_t(a)\|_1} W_t u_t(a)$$

$u_t(a)$ is a length-|T| binary vector, indicating the presence or absence of answer type.

$W_t \in \Re^{d_t \times |T|}$ is the parameter matrix

# Question Answering over Freebase with Multi-Column Convolutional Neural Networks [Dong et al., ACL 2015]

## Model Training

For every correct answer a of the question q, we randomly sample k wrong a’ from the set of candidate answers, and use them as the negative instances to estimate parameters.

$$l(q,a,a') = (m - S(q,a) + S(q,a'))_+$$

$$\min \sum_q \frac{1}{|A_q|} \sum_{a \in A_q} \sum_{a' \in R_q} l(q,a,a')$$

$$R_q \subseteq C_q \setminus A_q$$

$A_q$ is the correct answer set to question $q$.

$C_q$ is the set of canidate answer set to question $q$.

# KG-based Question/Answering

- Information Retrieval-based
  - Generate candidate answers
  - Ranking

- Semantic Parsing-based
  - Translate NLQ to logical forms
  - Executing

# Semantic Parsing

[Zettlemoyer et al., UAI 05]

Transforming natural language (NL) sentences into computer executable complete meaning representations (MRs) for domain-specic applications.

E.g., "Which states borders New Mexico ?"

Lambda-**calculus** [Alonzo Church, 1940 ]

$$\lambda x.state(x) \wedge borders(x, new\_mexico)$$

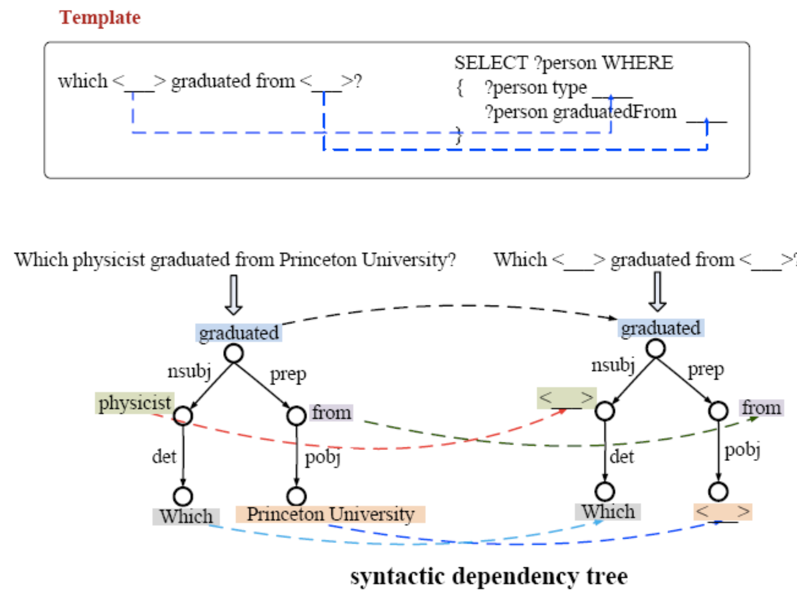
"**Simply typed Lambda-calculus** can express varies database query languages such as **relational algebra**, fixpoint logic and the complex object algebra." [Hillebrand et al., 1996]

# Semantic Parsing

- **Manually constructed rules**
  [Pedoe, AI magazine 2010]

- **Grammar-based, e.g.,**
  Combinatory Categorial Grammar
  [Zettlemoyer and Collins, UAI 2005]

- **Supervised Learning**
  [Berant and Liang, ACL 2014]

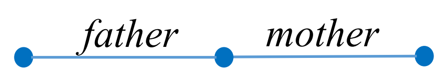



Template-based Approach [cite: Weiguo Zheng, Lei Zou, et al., SIGMOD 15]

# Challenges of Complex Questions

- **<u>Multi or Implicit relations</u>**
  - A complex question has multiple relations and some relations may be multi-hop or implicit.


"Which Russian astronauts died in the same place they were born in?"


"Who is the father of Trump's mother?"

# Challenges of Complex Questions

- **<u>Multi or No entities</u>**
  - A complex question may have multiple entities or do not have any entities.

“Which Russian astronauts were died in Moscow and born in Soviet Union”

“Who died in the same place they were born in?”

# Challenges of Complex Questions

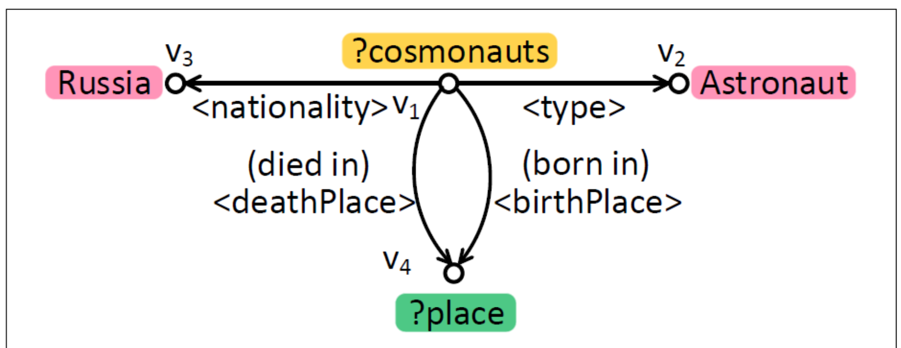
- **Variables and Co-reference**
  - Existing solutions only consider one variable: the answer
  - However: it can have multiple variables and some variables may refer to the same thing.

"Who died in the same place they were born in?"

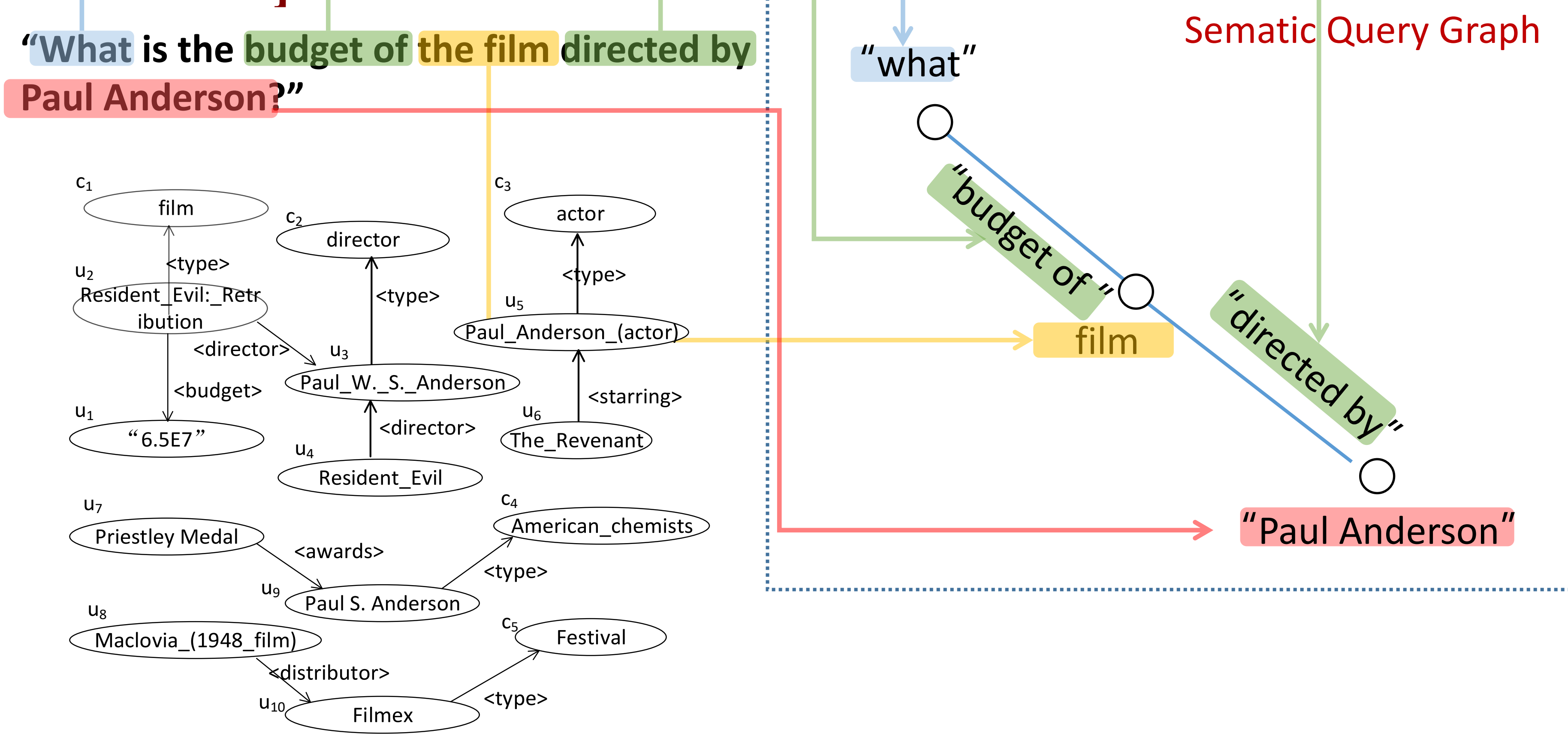
# Challenges of Complex Questions

- **<u>Composition</u>**
  - Simple Questions: one triple
  - Complex Questions: how to assemble multiple entities/variables/relations to logical forms or executable queries (query graphs)?

Which cosmonauts died in the same place they were born in?

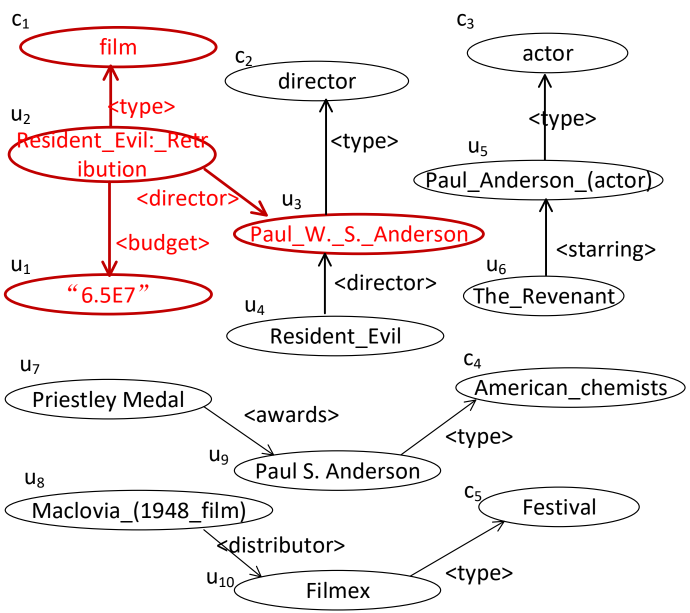


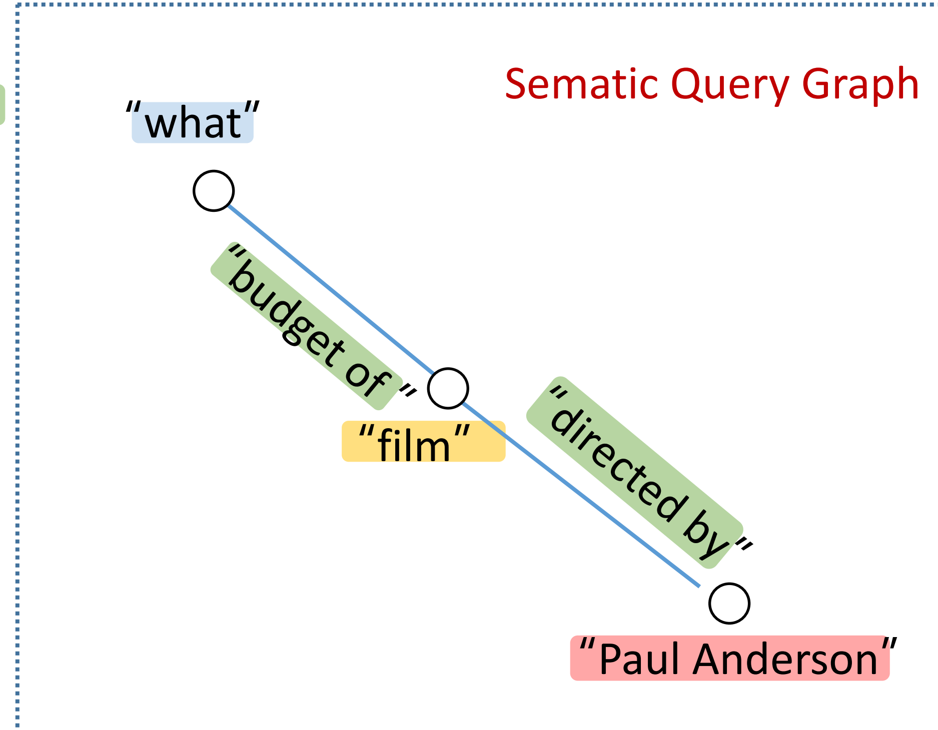
# Our Approach- Data Driven & Relation-first framework gAnswer [Zou et al, SIGMOD 14; Hu and Zou, et al., TKDE 17, EMNLP 18]



# Our Approach- Data Driven & Relation-first framework gAnswer [Zou et al, SIGMOD 14; Hu and Zou, et al., TKDE 17]

"What is the budget of the film directed by Paul Anderson?"

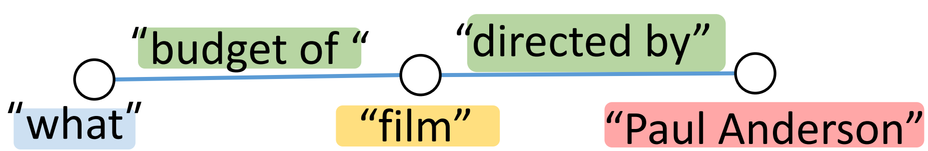


# Our Approach- gAnswer

**Semantic Query Graph:** A semantic query graph (denoted as $Q^S$) is a graph, in which each vertex $v_i$ is associated with an entity phrase, class phrase or wh-words in the question sentence $N$; and each edge $\overline{v_i v_j}$ is associated with a relation phrase in the question sentence $N$, $1 \leq i, j \leq |V(Q^S)|$

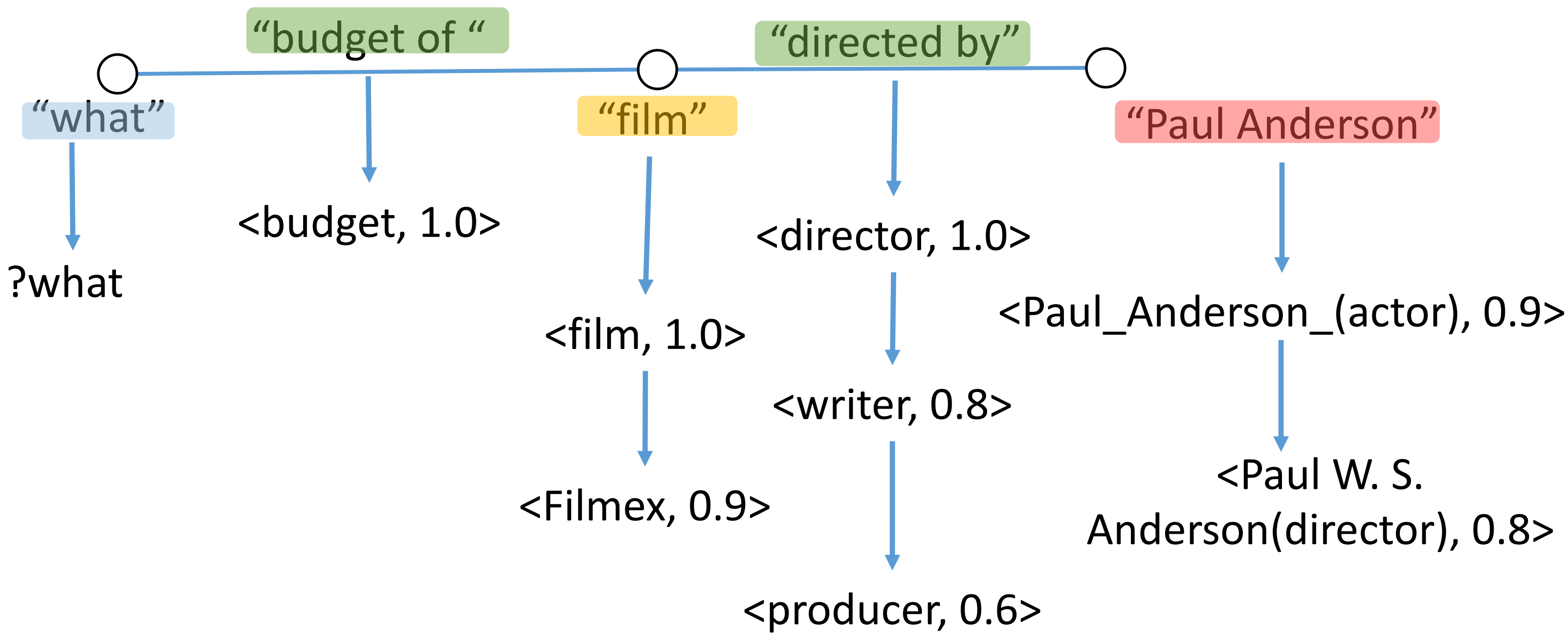
**"What is the budget of the film directed by Paul Anderson?"**



# Our Approach- gAnswer

Assume that we have built a SQG

- Entity Linking
- Relation Mapping



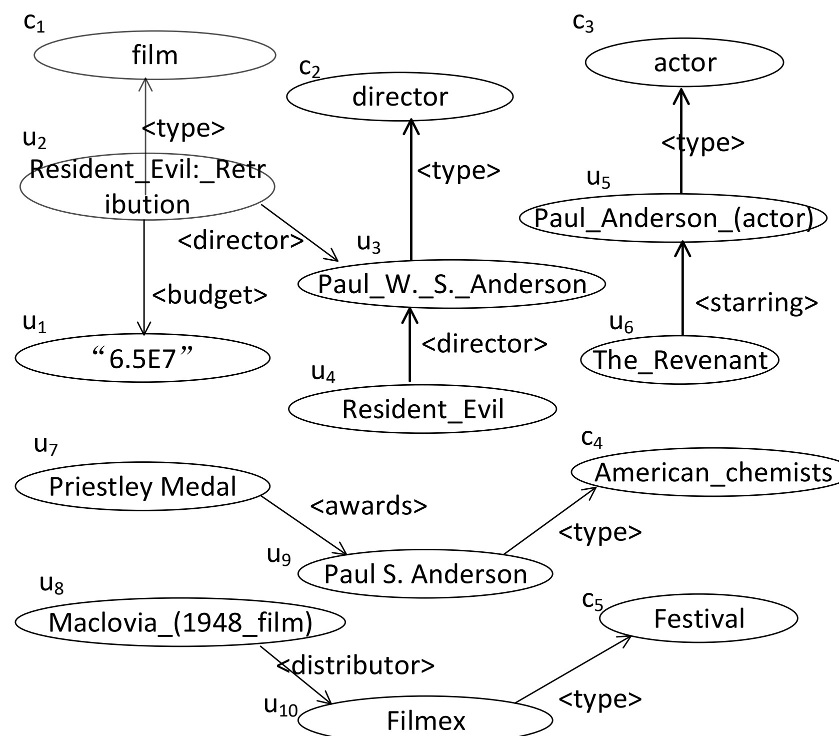
# Our Approach- Data Driven & Relation-first framework gAnswer [Zou et al, SIGMOD 14; Hu and Zou, et al., TKDE 17]

## Ambiguity

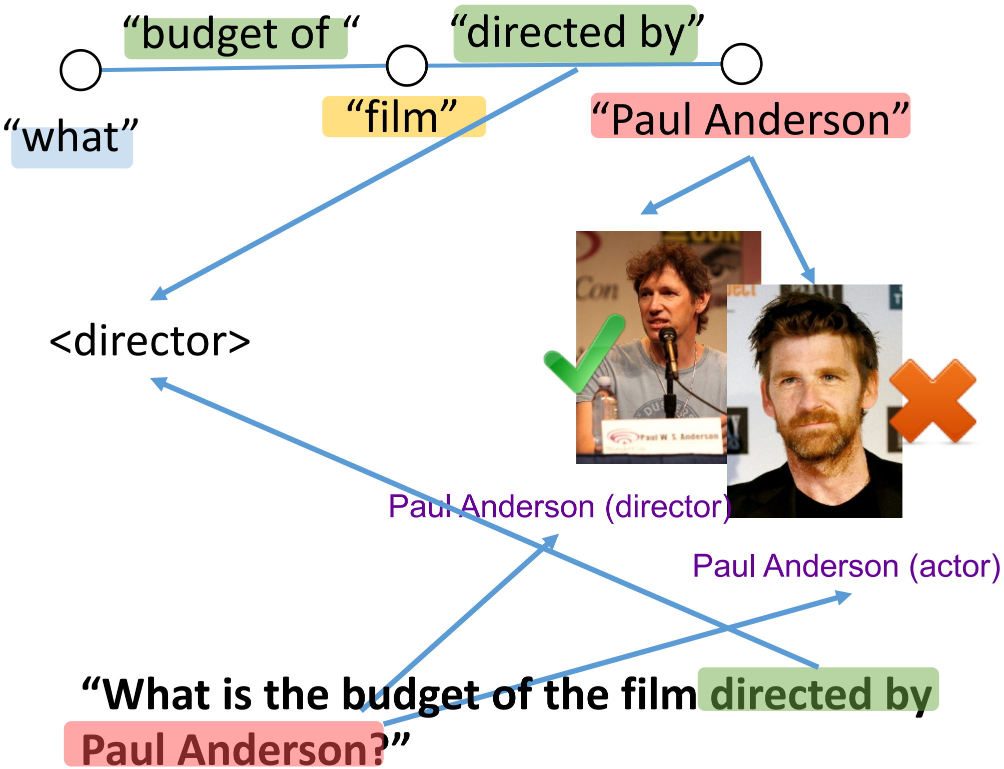
**"What is the budget of the film directed by Paul Anderson?"**

Paul W. S Anderson
(director)

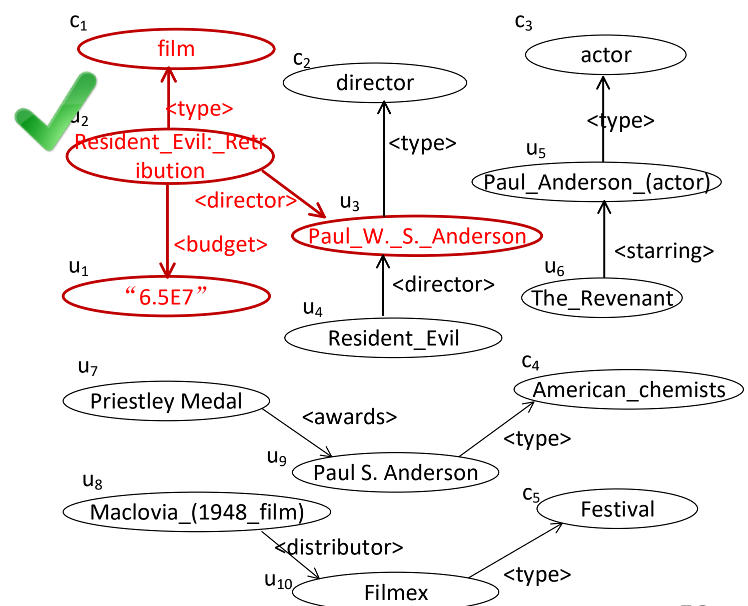
Paul S. Anderson
(actor)



# Our Approach- Data Driven & Relation-first framework gAnswer [Zou et al, SIGMOD 14; Hu and Zou, et al., TKDE 17]




## Graph Data Driven Disambiguation Approach



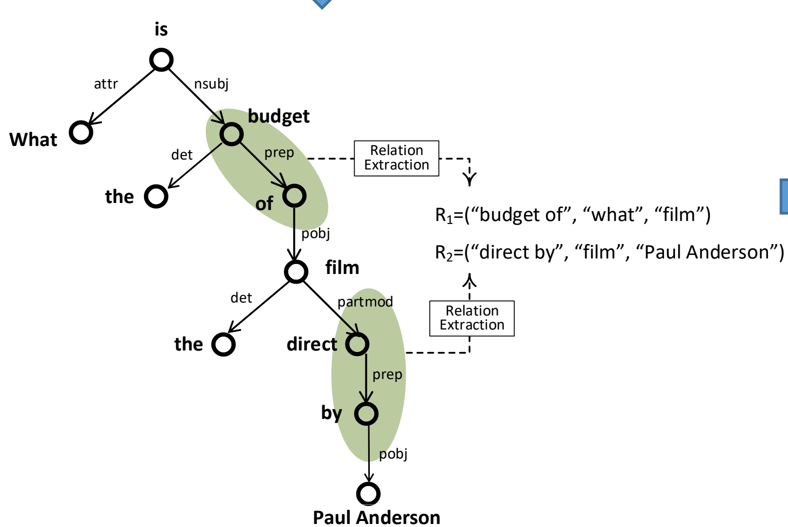
# Our Approach- gAnswer

Question: How to build a Semantic Query Graph ?

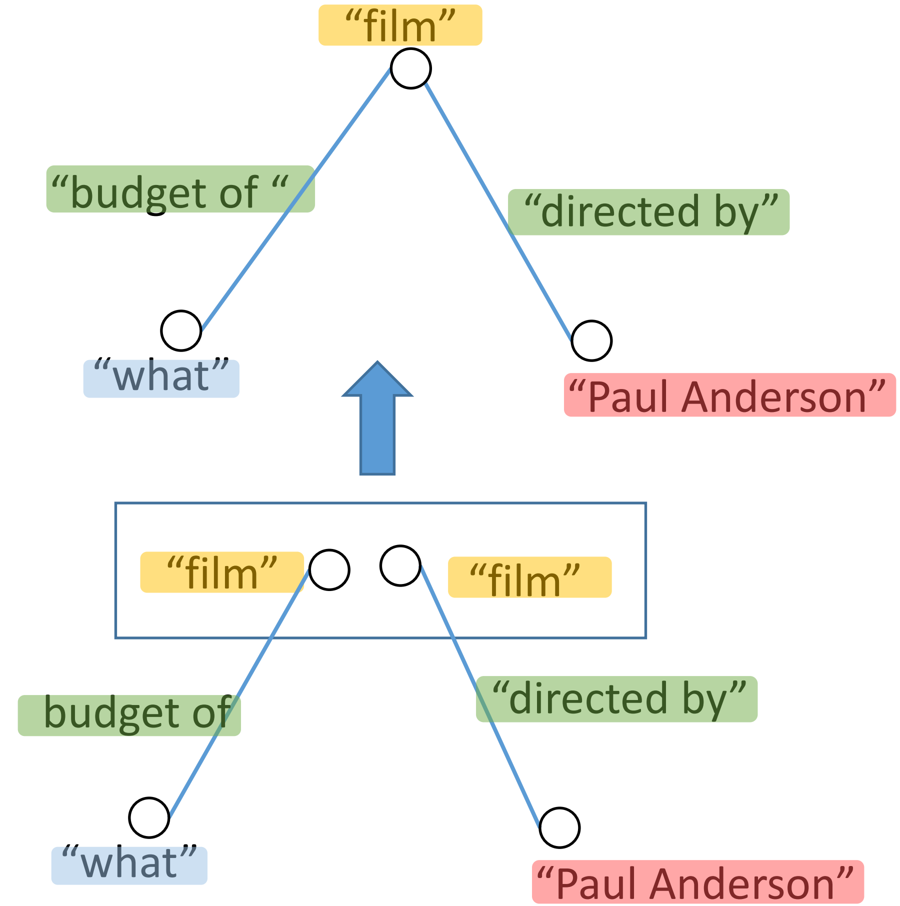
- Relation-First Approach [Zou et al, SIGMOD 14]
  - First find all relations mentioned in the question and model them as edges; Then, we assemble these edges together to form a SQG.
- Node-First Approach [Hu, Zou et al., TKDE 17]
  - First all mentioned entities/classes in the question and then link them to form a SQG.
- Learning-based Approach [Hu, Zou et al., EMNLP 18]
  - We propose a State Transition (ST) framework based on four primitive operations (expand, fold, connect and merge) to generate a SQG.

# gAnswer---Relation First

**"What is the budget of the film directed by Paul Anderson?"**

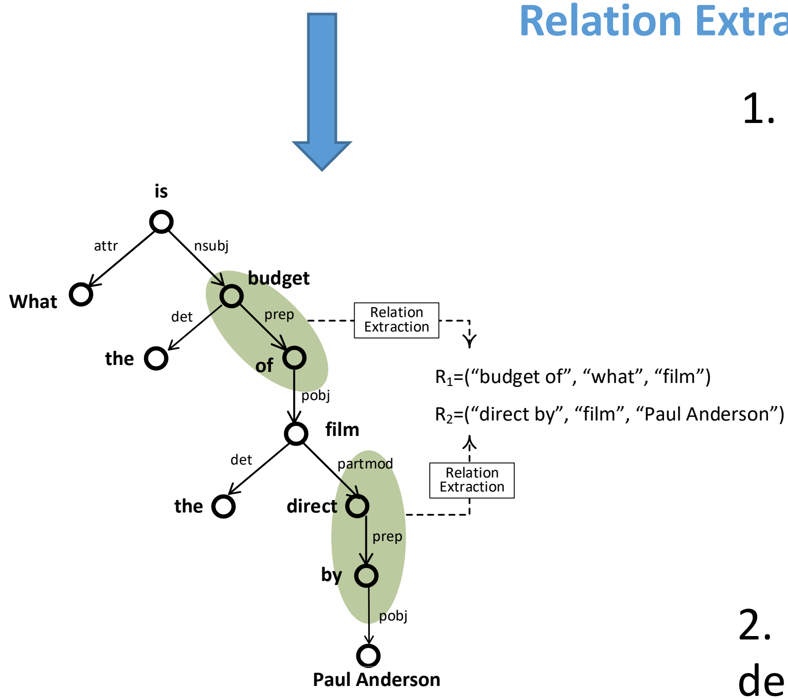



Dependency Parser Tree



# gAnswer---Relation First

**“What is the budget of the film directed by Paul Anderson?”**




Dependency Parser Tree

## Relation Extraction

1. Build a relation paraphrasing dictionary (Offline)

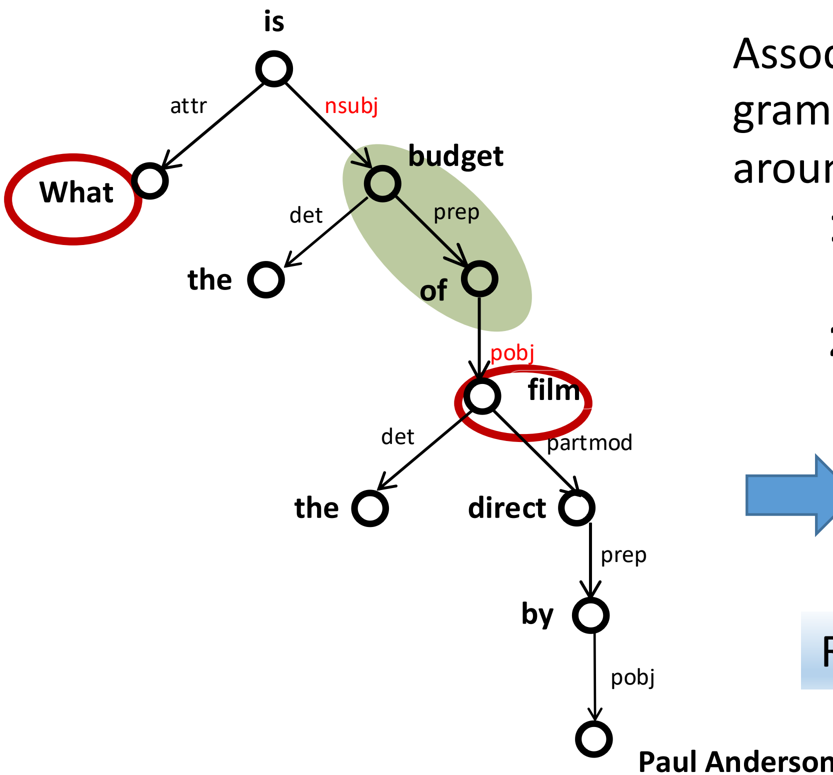
| Relation Mention | Predicate or Predicate Path | Confidence Probability |
|---|---|---|
| “directed by” | <director> | 1.0 |
| “starred by” | <starring> | 0.9 |
| “budget of” | <budget> | 0.8 |
| “uncle of” | <hasChild> <hasChild> <hasChild> | 0.8 |
| … … | … … | … … |

2. Finding all occurrences of relation phrases in the dependency parser tree (i.e., a connected subtree) (online)

# gAnswer---Relation First

**"What is the budget of the film directed by Paul Anderson?"**

## Finding Associated Nodes




Associated nodes are recognized also based on the grammatical subject-like and object-like relations around the relation phrases, which are listed as follow:

1. **subject-like relations**: subj, nsubj, nsubjpass, csubj, csubjpass, xsubj, poss, partmod;
2. **object-like relations**: obj, pobj, dobj, iobj

R1=( "budget of" , "what" , "film" )

R2=( "directed by" , "film" , "Paul Anderson" )

# gAnswer---Relation First

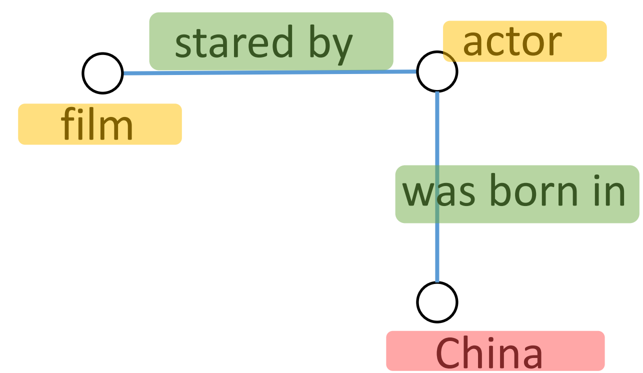
**Limitations of Relation First Approach**

cannot extract the relation between “*actor*” and “*Chinese*”

“show me all films started by a *Chinese actor*”

Implicit Relation

“show me all films started by an actor who was born in China”



# gAnswer---Relation First

**Limitations of Relation First Approach**

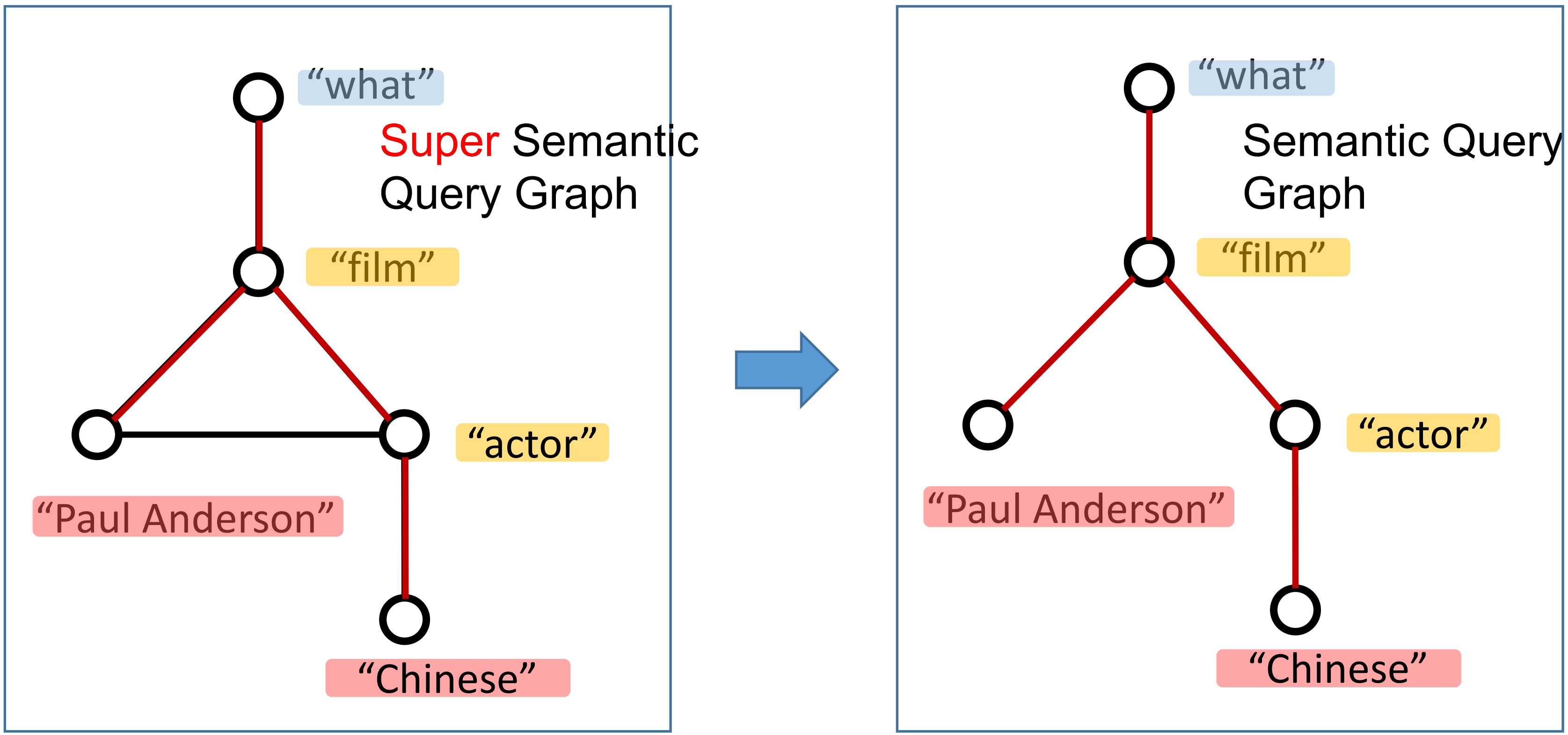
- SQG's structure highly relies on parser and heuristic rules
- RF approach assume that SQG's structure has no ambiguity.

It is not always true

A graph data driven approach for SQG's structure dis-ambiguity is desirable.

# gAnswer---Node First [Hu, Zou et al., TKDE 18]

**“What is the budget of the film directed by Paul Anderson and starred by a Chinese actor?”**

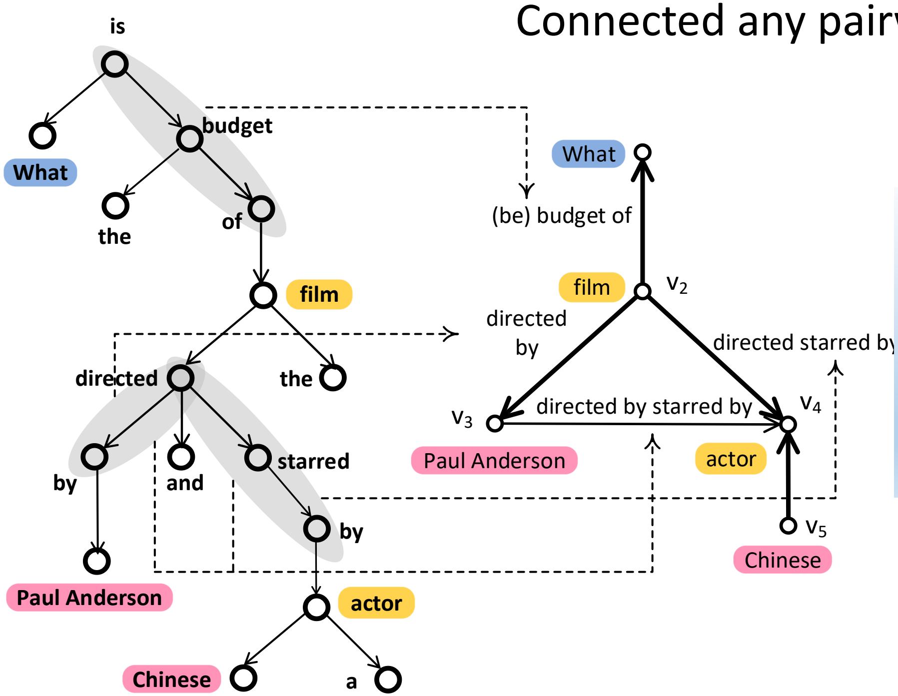


# gAnswer---Node First

How to build a Super Semantic Query Graph

**"What is the budget of the film directed by Paul Anderson and starred by a Chinese actor?"**

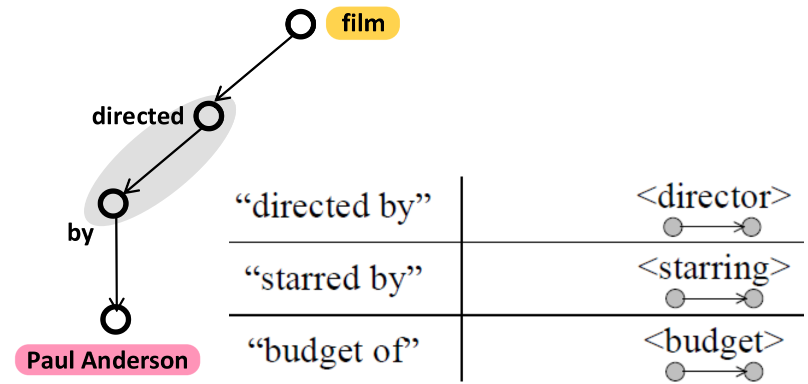
A Naïve Solution:
Connected any pairwise node → A complete graph ✗



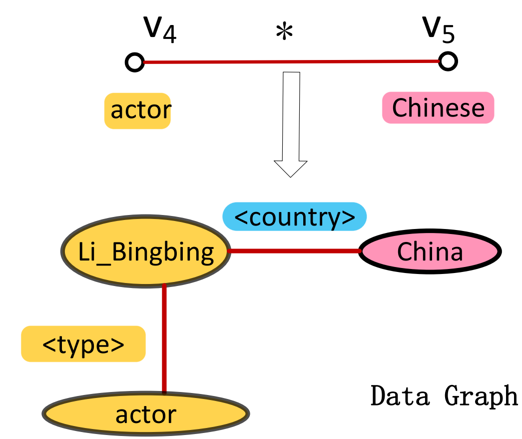

**Assumption 1.** Two nodes v1 and v2 has a semantic relation if and only if there exists no other node v* that occurs in the simple path between v1 and v2 of the dependency parse tree of question sentence N

# gAnswer---Node First

## Relation Mapping



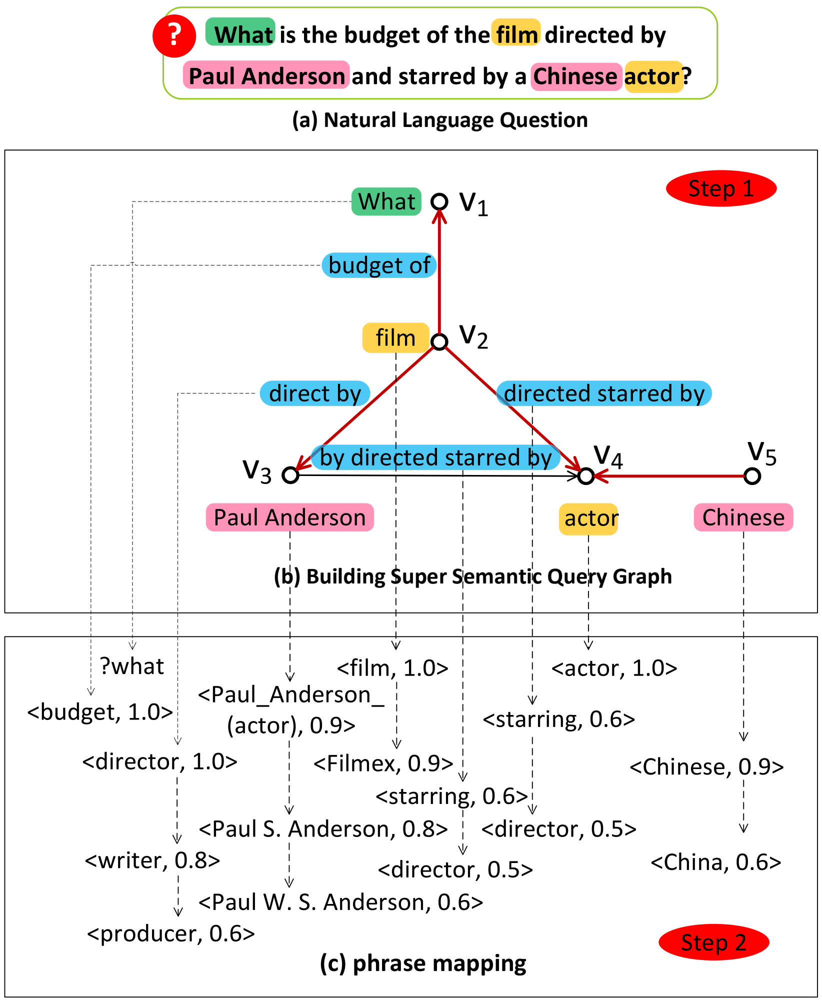

- Explicit Relation
    - using the paraphrasing dictionary– same as RF approach

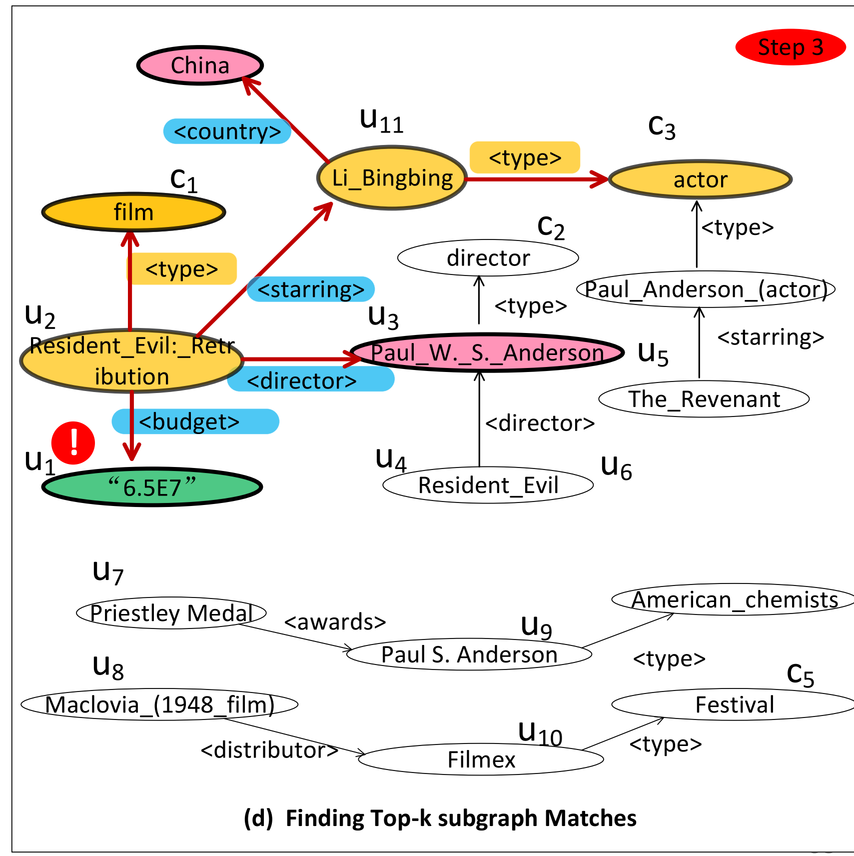



- Implicit Relation
    - Locating the two nodes in the data graph;
    - Finding the frequent predicate between them.

# gAnswer---Node First

## Approximate Match
## (allow dis-matching edges)



# Experiments

QALD is a series of evaluation campaigns on question answering over linked data.

TABLE 7
Evaluating QALD-6 Testing Questions (Total Question Number=100)

| | Processed | Right | Recall | Precision | F-1 |
|---|---|---|---|---|---|
| **NFF** | 100 | 68 | **0.70** | **0.89** | **0.78** |
| RFF | 100 | 40 | 0.43 | 0.77 | 0.55 |
| CANaLI | 100 | 83 | 0.89 | 0.89 | 0.89 |
| UTQA | 100 | 63 | 0.69 | 0.82 | 0.75 |
| KWGAnswer | 100 | 52 | 0.59 | 0.85 | 0.70 |
| SemGraphQA | 100 | 20 | 0.25 | 0.70 | 0.37 |
| UIQA1 | 44 | 21 | 0.63 | 0.54 | 0.25 |
| UIQA2 | 36 | 14 | 0.53 | 0.43 | 0.17 |
| DEANNA | 100 | 20 | 0.21 | 0.74 | 0.33 |
| Aqqu | 100 | 36 | 0.37 | 0.39 | 0.38 |

QALD-6 Competition Results

# Experiments

WebQuestions is widely used in Question Answering literatures and does not contain golden SPARQL queries.

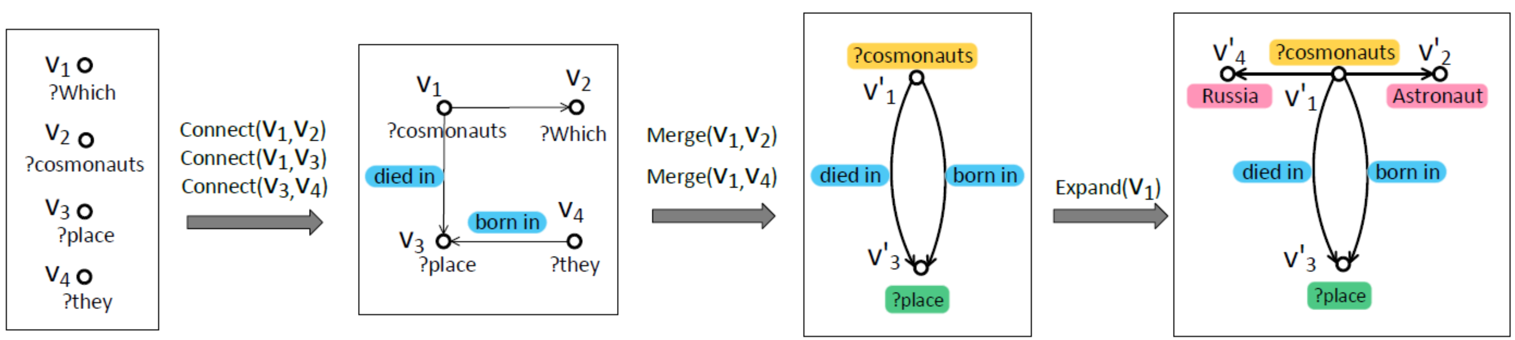
TABLE 8
Evaluating WebQuestions Testing Questions

| | Average F1 |
|---|---|
| **NFF** | 49.6% |
| RFF | 31.2% |
| Sempre | 35.7% |
| ParaSempre | 39.9% |
| Aqqu | 49.4% |
| STAGG | 52.5% |
| Yavuz et al. (2016) | **52.6%** |

WebQuestions Results

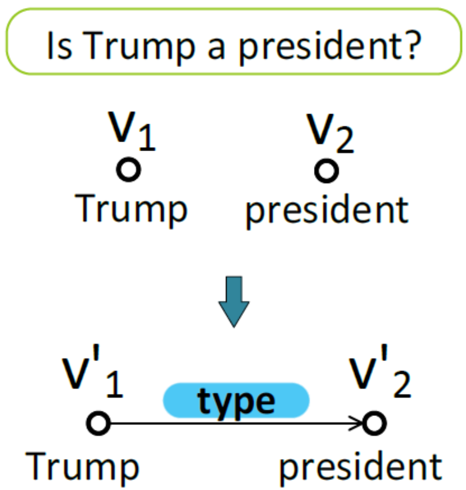
# gAnswer---State Transition (ST)-based Approach [Hu, Zou et al., EMNLP 18]

- **SQG Construction (State Transition)**
  - Initial state: those isolated nodes
  - Transition: by applying four primitive operations, led by a reward function
  - Final state: the semantic query graph (SQG), can be executed using algorithm in [Hu. et al, TKDE 2018]



# gAnswer---State Transition (ST)-based Approach [Hu, Zou et al., EMNLP 18]

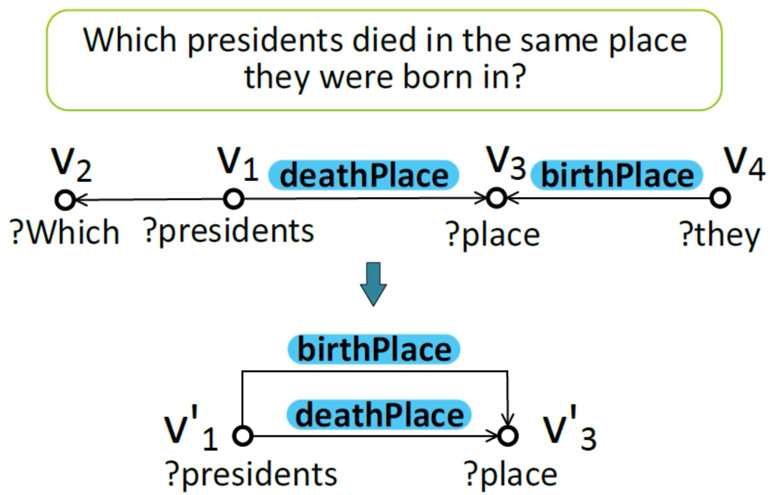
- **Connect Operation**
  - Given two operate nodes $u_1$ and $u_2$, we introduce an edge $\overline{u_1 u_2}$ between them by the connect operation.
  - The candidate relations (edge labels) of the edge $\overline{u_1 u_2}$ can be found through relation extraction model.



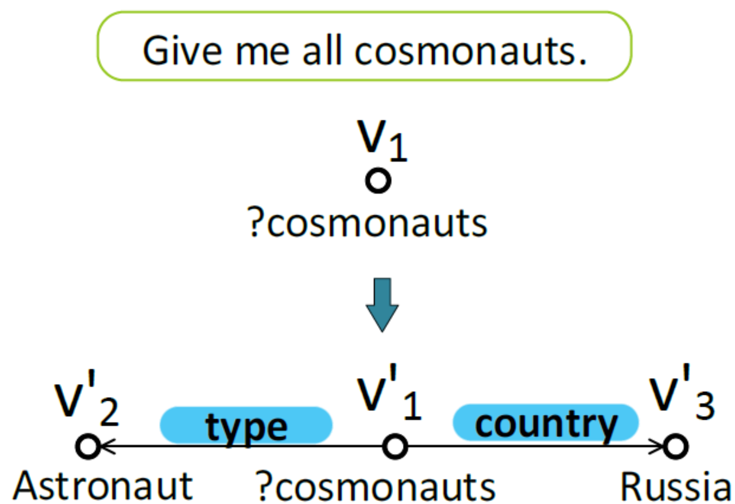
- **Merge Operation**
  - Given a SQG $Q^S$ = {*V*, *E*} and two operate nodes *u*, *v*, this operation is to merge the node *u* into *v*. The new SQG

$$Q'^S = \{V \setminus \{u\}, (E \setminus E^d) \cup E^a\}, E^d = \{\overline{uw} \in E\} \text{ , } E^a = \{\overline{vw} \mid \overline{uw} \in E \wedge w \neq v\}$$

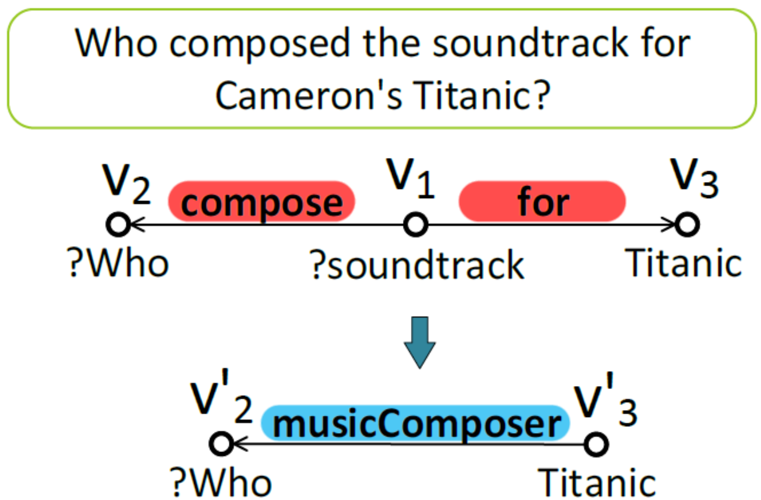
  - To support co-reference resolution.



- **Expand Operation**
  - Given a SQG $Q^S$ = {$V$, $E$} and the operate node $u \in V$, this operation is to expand $u$ to a subgraph $Q_u^S = \{V \cup V_u, E \cup E_u\}$.
  - To support nodes' hidden information.



- **<u>Fold Operation</u>**
  - To eliminate those nodes which are useless or mis-recognized



- State Transition
  - A greedy search algorithm similar with STAGG
  - Better intermediate states should have more opportunities to be visited and produce subsequent states.
  - Each state can be regarded as a partial SQG, and be estimated by a reward function (linear model)
  - To reduce the search space of the state transition process, we also propose several constraints for each operation.

- Reward Function
  - Taking the features and outputting the reward of corresponding state *s*
  - Features: such as confidence probability of nodes/relations, number of constant/variables, r

Given a SQG $Q^S = \{V, E\}$, the ranking score that we use to train our reward function $\gamma()$ is calculated by the following function.

$$R(Q^S) = \frac{|P(V)|}{|V'|} + \frac{|P(E)|}{|E'|} + \max(F(A_i, A'))$$

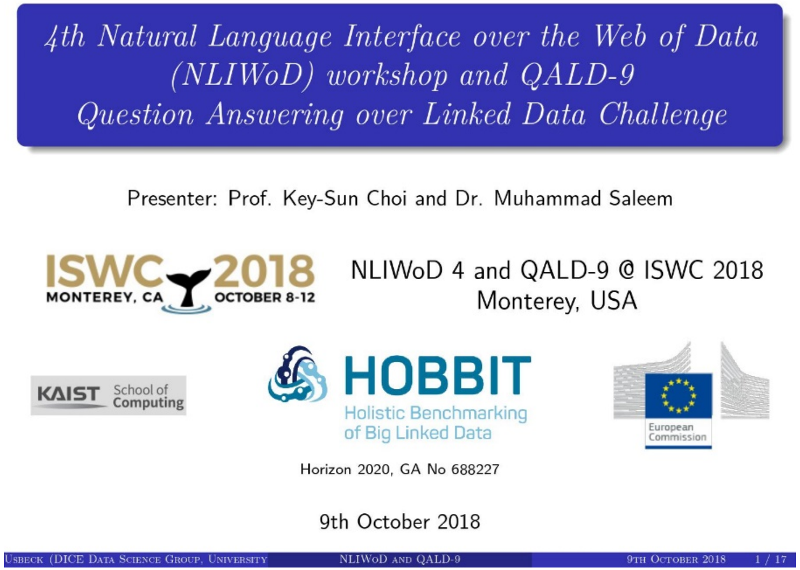
# Experiments

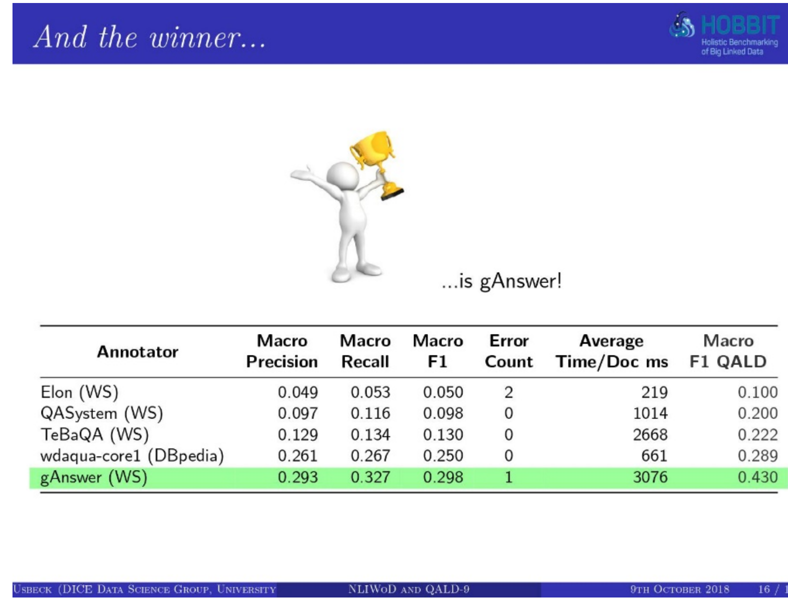
Table 2: The average F1 score of WebQuestions and ComplexQuestions benchmark

| | WQ | CQ |
|---|---|---|
| STF (Our approach) | **53.6%** | **54.3%** |
| STAGG (Yih et al., 2015) | 52.5% | - |
| QUINT (Abujabal et al., 2017) | 51.0% | 49.2% |
| NFF (Hu et al., 2018) | 49.6% | - |
| Aqqu (Bast and Haussmann, 2015) | 49.4% | 27.8% |
| Aqqu++ (Bast and Haussmann, 2015) | 49.4% | 46.7% |

Table 3: Evaluating QALD-6 Testing Questions

| | Processed | Right | Recall | Precision | F-1 |
|---|---|---|---|---|---|
| Our approach | 100 | 70 | 0.72 | 0.89 | 0.80 |
| CANaLI | 100 | 83 | 0.89 | 0.89 | **0.89** |
| UTQA | 100 | 63 | 0.69 | 0.82 | 0.75 |
| KWGAnswer | 100 | 52 | 0.59 | 0.85 | 0.70 |
| SemGraphQA | 100 | 20 | 0.25 | 0.70 | 0.37 |
| UIQA1 | 44 | 21 | 0.63 | 0.54 | 0.25 |
| UIQA2 | 36 | 14 | 0.53 | 0.43 | 0.17 |
| NFF | 100 | 68 | 0.70 | 0.89 | 0.78 |
| gAnswer | 100 | 40 | 0.43 | 0.77 | 0.55 |
| Aqqu | 100 | 36 | 0.37 | 0.39 | 0.38 |



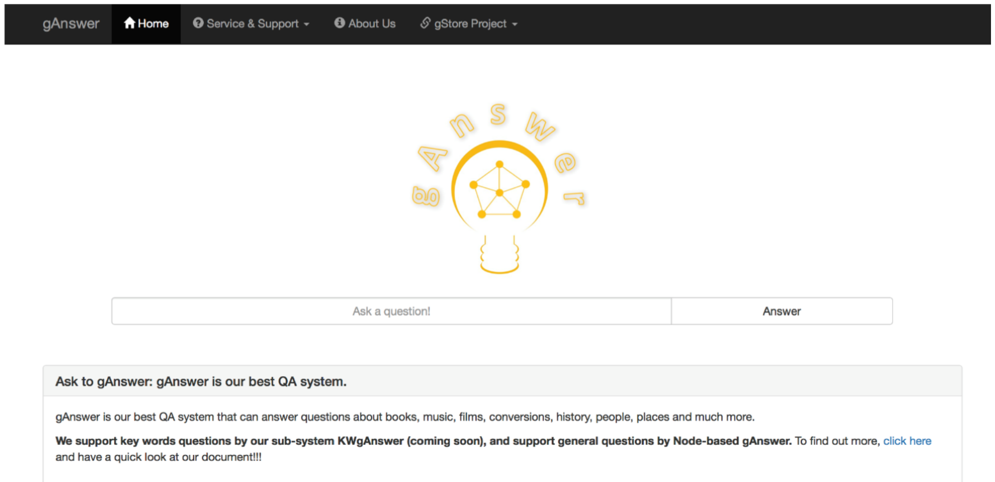




| Annotator | Macro Precision | Macro Recall | Macro F1 | Error Count | Average Time/Doc ms | Macro F1 QALD |
|---|---|---|---|---|---|---|
| Elon (WS) | 0.049 | 0.053 | 0.050 | 2 | 219 | 0.100 |
| QASystem (WS) | 0.097 | 0.116 | 0.098 | 0 | 1014 | 0.200 |
| TeBaQA (WS) | 0.129 | 0.134 | 0.130 | 0 | 2668 | 0.222 |
| wdaqua-core1 (DBpedia) | 0.261 | 0.267 | 0.250 | 0 | 661 | 0.289 |
| gAnswer (WS) | 0.293 | 0.327 | 0.298 | 1 | 3076 | 0.430 |

gAnswer won the first place of QALD-9

# gAnswer

Online Demo: URL: http://ganswer.gstore-pku.com/




Github Project: https://github.com/pkumod/gAnswer

# Is it Possible ?

Semantic Parsing（NLP）+Query Evaluation （DB）

Where is the nearest post office ?

$$\arg\min(\lambda x.POST(x) \wedge dis(HERE, x))$$

**SPARQL**

```
SELECT ?x WHERE {
?x rdf:type Post.
?x :longitude ?o.
?x :latitude ?a. }
ORDERY BY Dist(HERE, [?o, ?a])
LIMIT 1
```

# gAnswer---related publication.

- Sen Hu, Lei Zou, Xinbo Zhang, A State-transition Framework to Answer Complex Questions over Knowledge Base, **EMNLP** 2018
- Xinbo Zhang, Lei Zou: IMPROVE-QA: An Interactive Mechanism for RDF Question/Answering Systems. **SIGMOD** Conference 2018: 1753-1756 (Demo)
- Weiguo Zheng, Jeffrey Xu Yu, Lei Zou, Hong Cheng: Question Answering Over Knowledge Graphs: Question Understanding Via Template Decomposition. **PVLDB** 11(11): 1373-1386 (2018)
- Shuo Han, Lei Zou, Jeffrey Xu Yu, Dongyan Zhao: Keyword Search on RDF Graphs - A Query Graph Assembly Approach. **CIKM** 2017: 227-236
- Sen Hu, Lei Zou, Haixun Wang, Jeffrey Xu Yu, Wenqiang He: Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs. IEEE **TKDE** 2017
- Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, Dongyan Zhao: Natural language question answering over RDF: a graph data driven approach. SIGMOD Conference 2014: 313-324
- Ruizhe Huang, Lei Zou: Natural language question answering over RDF data. **SIGMOD** Conference 2013: 1289-1290 (undergraudate student's poster)
- Weiguo Zheng, Lei Zou, Xiang Lian, Jeffrey Xu Yu, Shaoxu Song, Dongyan Zhao. How to Build Templates for RDF Question/Answering: An Uncertain Graph Similarity Join Approach **SIGMOD** Conference, 2015.

References：


- Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, Oksana Yakhnenko: Translating Embeddings for Modeling Multi-relational Data. NIPS 2013: 2787-2795
- Luke S. Zettlemoyer, Michael Collins: Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. UAI 2005: 658-666
- Pablo N. Mendes, Max Jakob, Christian Bizer: DBpedia: A Multilingual Cross-domain Knowledge Base. LREC 2012: 1813-1817
- Fabian M. Suchanek, Gjergji Kasneci and Gerhard Weikum, Yago - A Core of Semantic Knowledge, 16th international World Wide Web conference (WWW 2007)
- Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor: Freebase: a collaboratively created graph database for structuring human knowledge. SIGMOD Conference 2008: 1247-1250
- Peter Buneman, Gao Cong, Wenfei Fan, Anastasios Kementsietsidis: Using Partial Evaluation in Distributed Query Evaluation. VLDB 2006: 211-222
- Yuk Wah Wong, Raymond J. Mooney: Learning for Semantic Parsing with Statistical Machine Translation. HLT-NAACL 2006
- C. Unger, L. Bühmann, J. Lehmann, A.-C. N. Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over RDF data. In WWW, pages 639–648, 2012
- Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He and Dongyan Zhao, Natural Language Question Answering over RDF ---- A Graph Data Driven Approach , SIGMOD (2014)
- Lei Zou, Jinghui Mo, Lei Chen,M. Tamer Özsu, Dongyan Zhao, gStore: Answering SPARQL Queries Via Subgraph Matching, in Proceedings of 37th International Conference on Very Large Databases (VLDB), 2011.
- Peng Peng, Lei Zou, Tamer Ozsu, Lei Chen, Dongyan Zhao, Processing SPARQL queries over distributed RDF graphs, accepted by VLDB Journal
- Church, A. “A Formulation of the Simple Theory of Types”. Journal of Symbolic Logic 5: 1940. doi:10.2307/2266170

References ：


- C. Unger, L. Bühmann, J. Lehmann, A.-C. N. Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over RDF data. In WWW, pages 639–648, 2012
- Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He and Dongyan Zhao, Natural Language Question Answering over RDF ---- A Graph Data Driven Approach , SIGMOD (2014)
- Lei Zou, Jinghui Mo, Lei Chen,M. Tamer Özsu, Dongyan Zhao, gStore: Answering SPARQL Queries Via Subgraph Matching, in Proceedings of 37th International Conference on Very Large Databases (VLDB), 2011.
- Antoine Bordes, Sumit Chopra, Jason Weston: Question Answering with Subgraph Embeddings. EMNLP 2014: 615-620
- William Tunstall-Pedoe: True Knowledge: Open-Domain Question Answering using Structured Knowledge and Inference. AI Magazine 31(3): 80-92 (2010)
- Luke S. Zettlemoyer, Michael Collins: Learning Context-Dependent Mappings from Sentences to Logical Form. ACL/IJCNLP 2009: 976-984
- Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL’14), Baltimore, USA
- Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, Dongyan Zhao: Question Answering on Freebase via Relation Extraction and Textual Evidence. ACL (1) 2016
- Sen Hu, Lei Zou, Jeffrey Xu Yu, Haixun Wang, Dongyan Zhao, Answering Natural Language Questions by Subgraph Matching over Knowledge Graphs, accepted by IEEE Transactions on Knowledge and Data Engineering, 2017
- Wen-tau Yih, Ming-Wei Chang, Xiaodong He, Jianfeng Gao: Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. ACL (1) 2015: 1321-1331
- Li Dong, Mirella Lapata: Language to Logical Form with Neural Attention. ACL (1) 2016

# Thanks！

zoulei@pku.edu.cn