# SPARQA: Skeleton-based Semantic Parsing for Complex Questions over Knowledge Bases

Yawei Sun (**ywsun@smail.nju.edu.cn**), Lingling Zhang, Gong Cheng, Yuzhong Qu
National Key Laboratory for Novel Software Technology, Nanjing University, China

Code: https://github.com/nju-websoft/SPARQA

## Task and Approach Overview

### Task

- Answering complex questions (multiple predicates) over knowledge bases.
- Example: What movie that Miley Cyrus *acted in* had a *director* named Tom Vaughan ?

### Challenges

- Syntactic parsing error of long complex question
- Structural heterogeneity between question and KB

### Contributions

- Contribution 1: skeleton parsing

We propose a skeleton grammar to represent the high-level structure of a complex question. This lightweight formalism and our parsing algorithm help to improve the accuracy of the downstream semantic parsing.

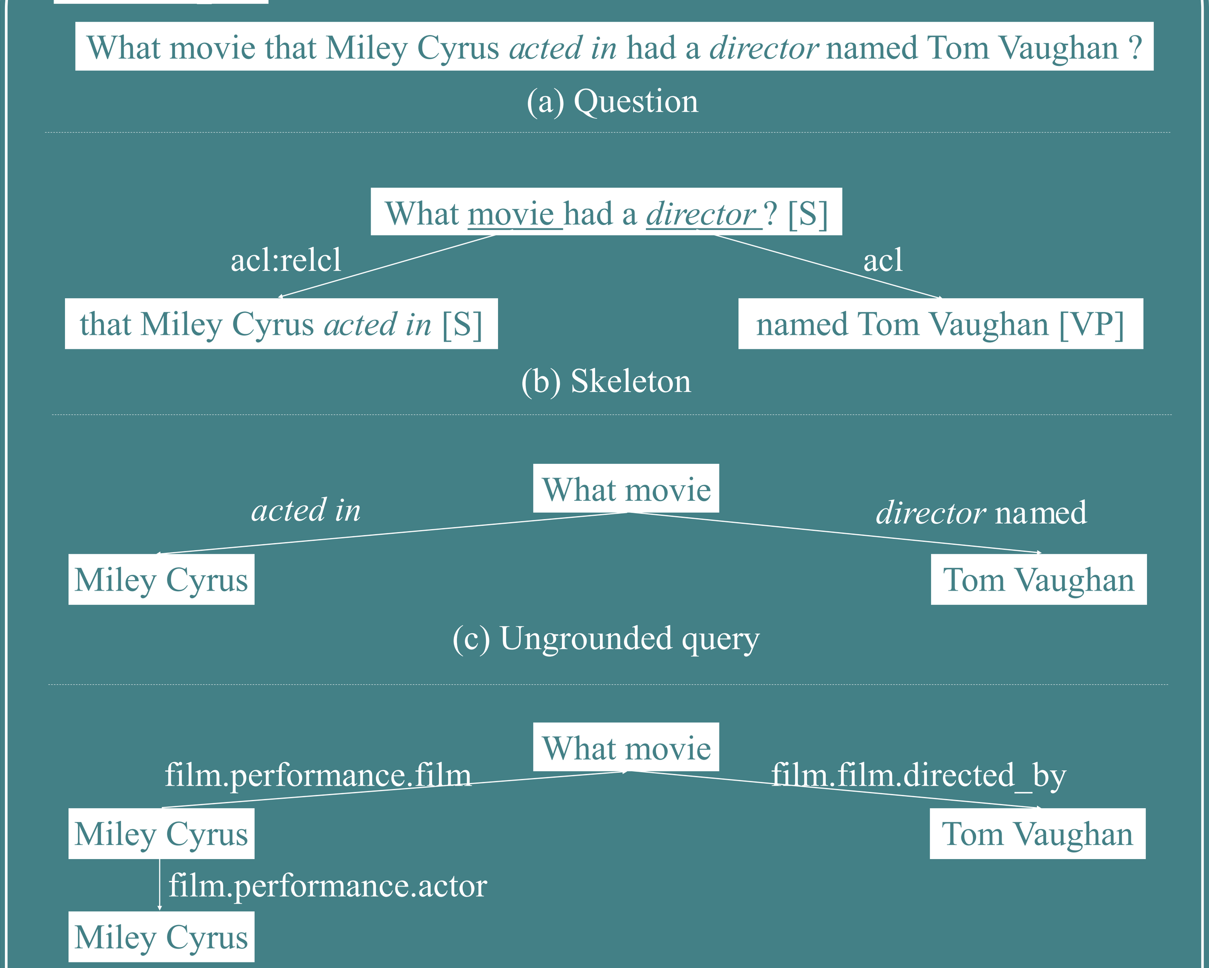
- Contribution 2: skeleton annotation

To train and evaluate our algorithm for skeleton parsing, we manually annotate the skeleton structure for over 10K questions in two KBQA datasets. We make this resource public to support future research (https://github.com/nju-websoft/SPARQA).

- Contribution 3: multi-strategy scoring

We combine sentence-level and word-level scoring to rank grounded queries. The former mines and matches sentence patterns. The latter processes words and trains a novel neural model to compute similarity.

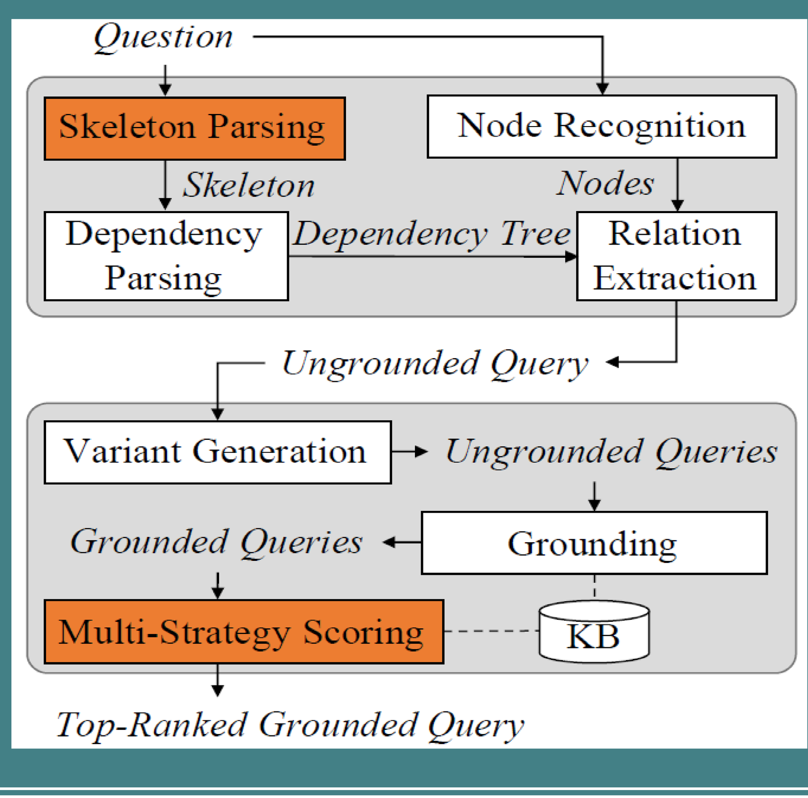
### Example




### Overview

**SPARQA** is a *S*keleton-based semantic *PAR*sing for *Q*uestion *A*nswering. SPARQA consists of two steps:

- (1) Ungrounded query generation: generate a KB-independent ungrounded query.
- (2) Grounded query generation: generate a KB-dependent grounded query.




## Approach Details

### (1) Ungrounded Query Generation

**Input**: Question

**Output**: KB-independent ungrounded query

**Solution**: (i) skeleton parsing, (ii) dependency parsing, (iii) node recognition, and (iv) relation extraction

(i) Skeleton Parsing

- Propose a BERT-based parsing algorithm (see below algorithm 1) to identify high-level skeleton structure of long complex question.

Contribution 1: skeleton parsing

(ii) Dependency Parsing

- Run standard dependency parsing (Stanford CoreNLP) of each text span to fine-grained semantic parsing.

(iii) Node Recognition

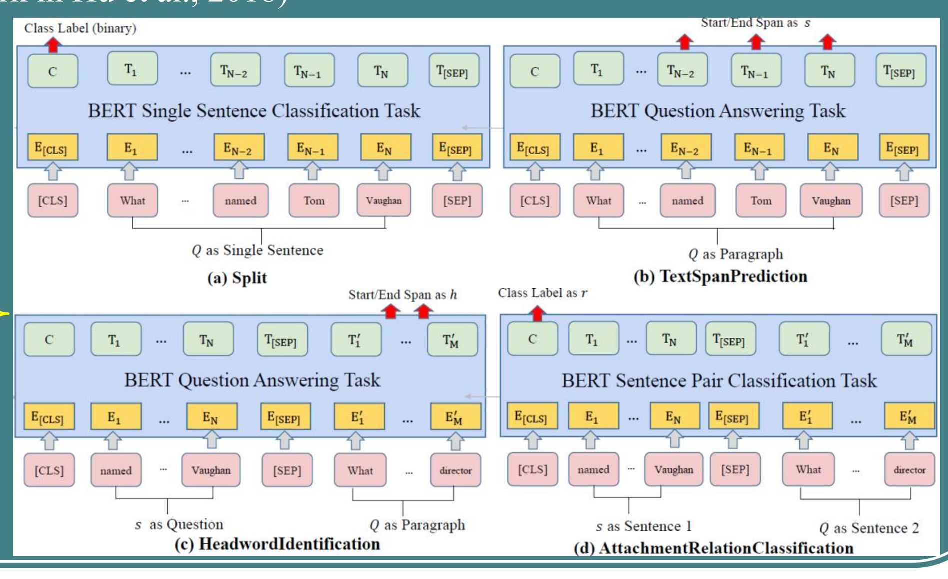
- Use a combination of Stanford's NER, SUTime, and a BERT-based token classifier to recognize nodes of ungrounded query (entity mention, class mention, literal).

(iv) Relation Extraction

- Use an existing method (node-first framework in Hu et al., 2018) to extract relation of ungrounded query

**Algorithm 1** Skeleton Parsing

**Require:** A sentence $Q$
**Ensure:** The skeleton of $Q$
  $T \leftarrow$ tree with a root node $Q$
  **while** Split($Q$) is true **do**
    $s \leftarrow$ TextSpanPrediction($Q$)
    $h \leftarrow$ HeadwordIdentification($s$, $Q$)
    $r \leftarrow$ AttachmentRelationClassification($s$, $Q$)
    Remove $s$ from $Q$
    Grow $T$ with relation $r$ from $h \in Q$ to $s$
  **end while**
  **return** $T$




### (2) Grounded Query Generation

**Input**: KB-independent ungrounded query

**Output**: top-ranked grounded query

**Solution**: (i) variant generation, (ii) grounding, and (iii) multi-strategy scoring

(i) Variant Generation

- Generate a set of structural variants of ungrounded query by contracting edge between class nodes and/or subdividing an edge with an inserted mediator node to address structural heterogeneity.

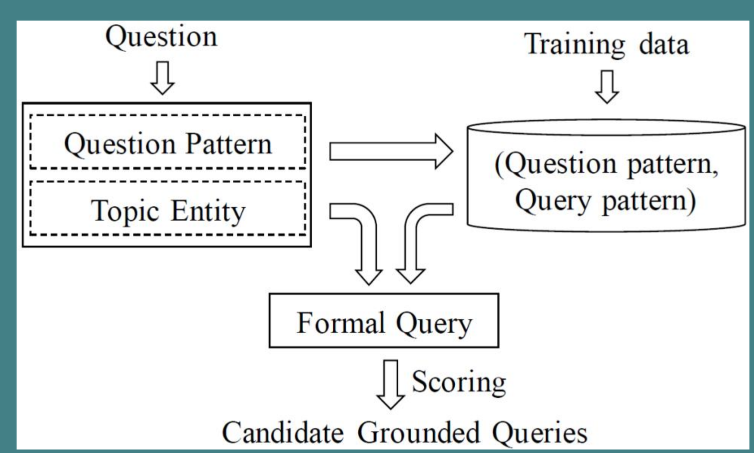
(ii) Grounding

- Link each node to an entity, class in the KB by using a dictionary compiled from ClueWeb and KB-specific resources.
- Enumerate all possible predicates that connect adjacent nodes to generate candidate formal grounded queries.

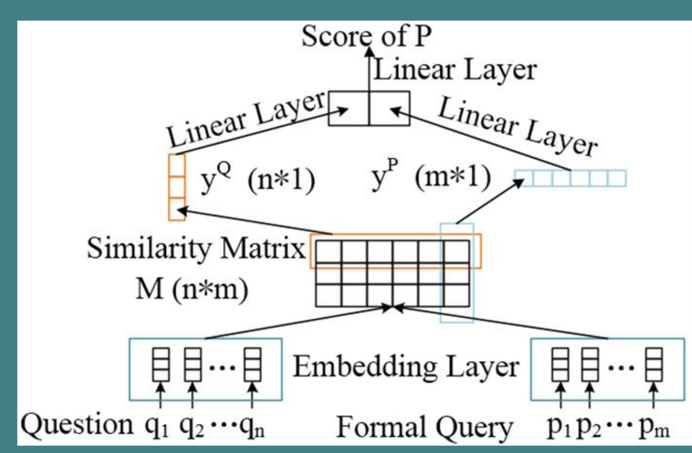
(iii) Multi-Strategy Scoring

Contribution 3: multi-strategy scoring

- Propose a multi-strategy method to rank candidate formal grounded queries.
  - Sentence-level scorer: mine and match sentence/query pattern to score candidates queries (see the left figure).
  - Word-level scorer: propose a novel word-level neural model to compute similarity (see the right figure).




Sentence-level Scorer




Word-level Scorer

## Experiments

### Evaluation Design

**Datasets**

- GraphQuestions : 5,166 Questions
- ComplexWebQuestions (v1.1): 34,689 Questions

**Skeleton Annotation**

- Skeleton annotation for 10K questions in the datasets

Contribution 2: skeleton annotation

**SOTA Methods**

- For GraphQuestions
  - SEMPRE (Berant et al., 2013)
  - PARASEMPRE (Berant and Liang 2014)
  - JACANA (Yao and Durme 2014)
  - UDEPLAMBDA (Reddy et al., 2017)
  - SCANNER (Cheng et al., 2017)
  - PARA4QA (Dong et al., 2017)
- For ComplexWebQuestions
  - MHQA-GRN (Song et al., 2018)
  - SIMPQA+ PRETRAINED (Talmor and Berant 2018b)
  - SPLITQA+PRETRAINED (Talmor and Berant 2018a)
  - SPLITQA+data augmentation (Talmor and Berant 2018a)
  - PullNet (Sun, Bedrax-Weiss, and Cohen 2019)

**Results on GraphQuestions**

| | F1 |
|---|---|
| SEMPRE | 10.08 |
| PARASEMPRE | 12.79 |
| JACANA | 5.08 |
| UDEPLAMBDA | 17.70 |
| SCANNER | 17.02 |
| PARA4QA | 20.40 |
| SPARQA | 21.53 |

**Results on ComplexWebQuestions**

| | P@1 |
|---|---|
| MHQA-GRN | 30.10 |
| SIMPQA+ PRETRAINED | 19.90 |
| SPLITQA+PRETRAINED | 25.90 |
| SPLITQA+data augmentation | 34.20 |
| PullNet | 45.90 |
| SPARQA | 31.57 |

**Ablation study on ComplexWebQuestions**

| | P@1 |
|---|---|
| SPARQA | 31.57 |
| SPARQA w/o skeleton parsing | 29.39 |
| SPARQA w/o sentence-level scorer | 26.45 |
| SPARQA w/o word-level scorer | 26.11 |

**Skeleton intrinsic evaluation on 1,000 test questions in ComplexWebQuestion**

| | |
|---|---|
| Split (ACC) | 99.42 |
| TextSpanPrediction (ACC) | 97.17 |
| HeadwordIdentification (ACC) | 97.22 |
| AttachmentRelationClassification (ACC) | 99.14 |
| Skeleton Overall (LAS) | 93.73 |

### Error Analysis

- Node recognition and linking (37%)
- Skeleton parsing (5%)

  Long-distance attachment is sometime not found. e.g., What country speaks Germanic languages *with a capital called Brussels* ?
- Ungrounded query (10%)
- Structural heterogeneity (22%)

  Ungrounded query is a Path-structured, but grounded query is a star-structured. e.g., Who is the *prime minister* of the country that has Loma ?
- Candidate queries scoring (15%)
- Others (11%)

### Take-home Message

- SPARQA shows that coarse-grained skeleton parsing can help to improve the accuracy of the downstream fine-grained semantic parsing.
- Code: https://github.com/nju-websoft/SPARQA