

RnG-KBQA: Generation Augmented Iterative Ranking for Knowledge Base Question Answering



Xi Ye♠ Semih Yavuz♦ Kazuma Hashimoto♦
Yingbo Zhou♦ Caiming Xiong♦

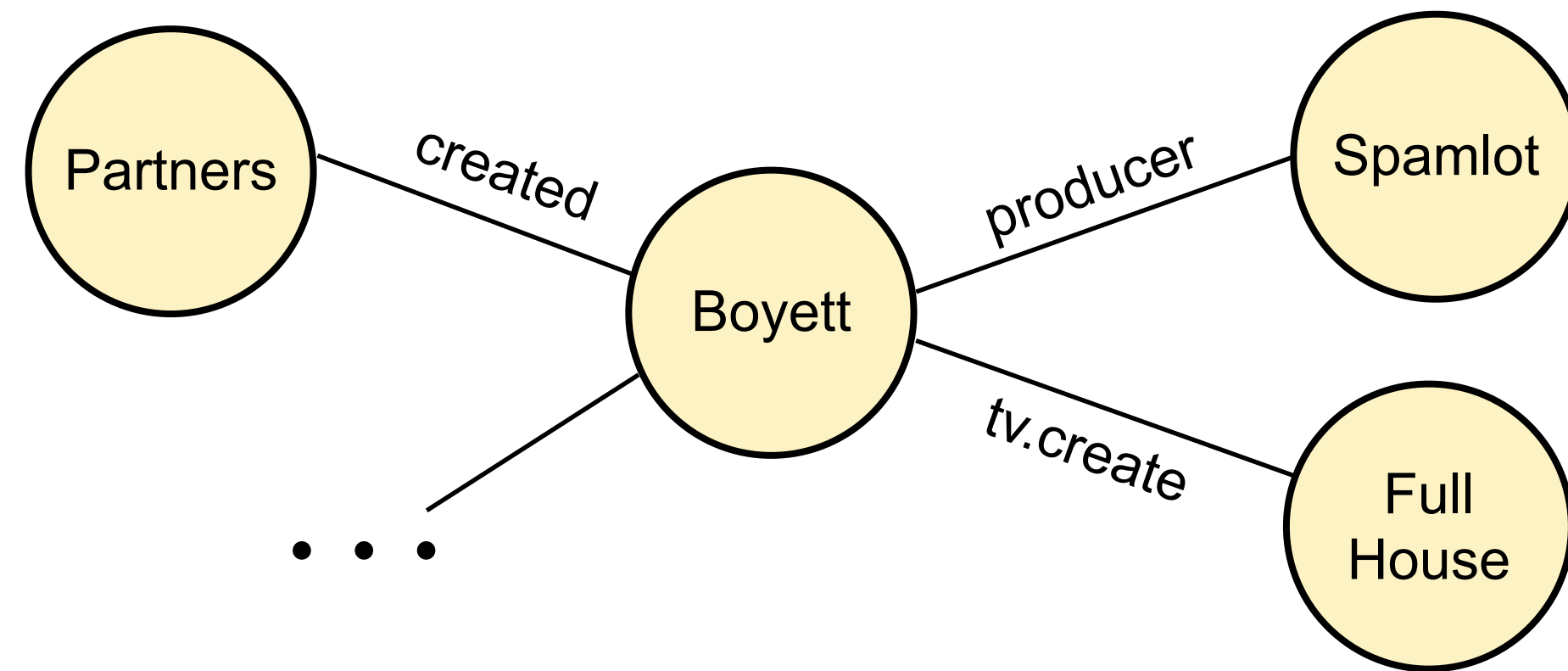
♠ *The University of Texas at Austin*
♦ *Salesforce Research*





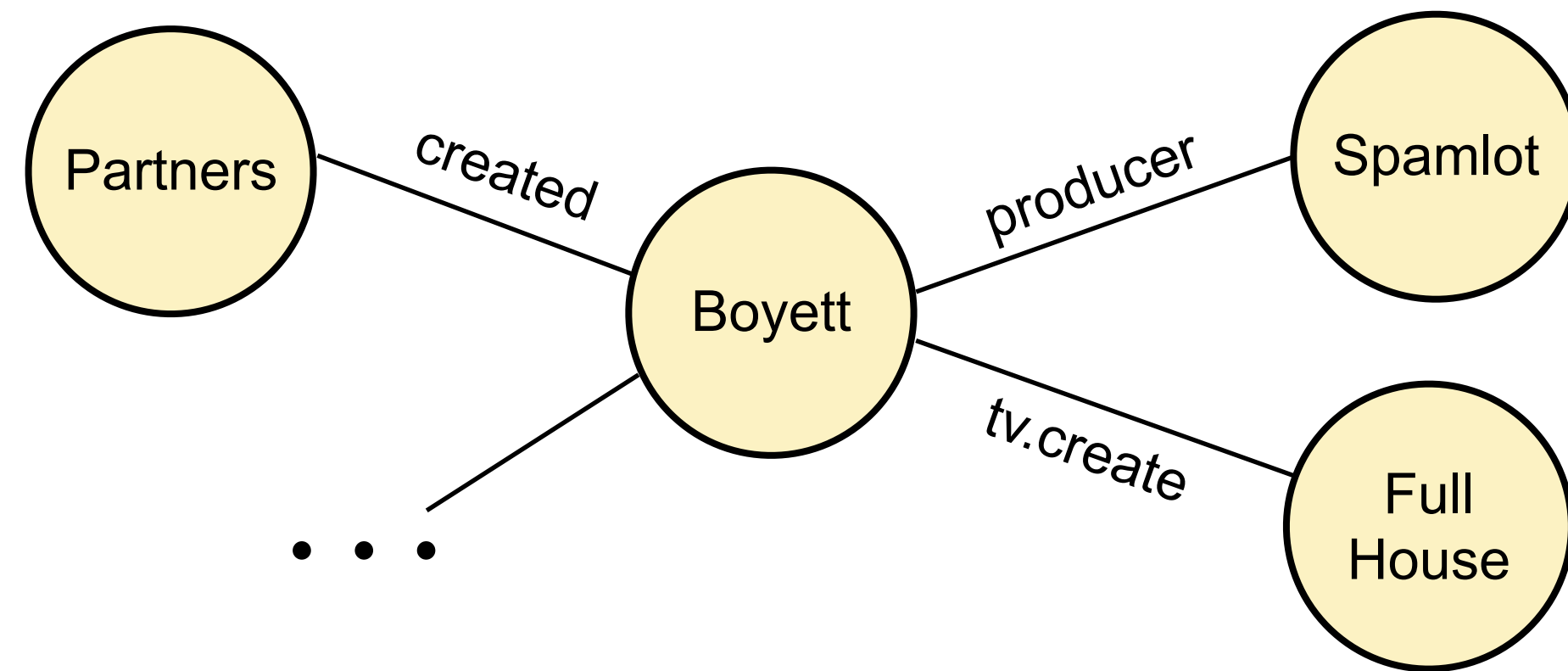
Question Answering over KB

Question Answering over KB



- ▶ KBs are reliable sources of knowledge, but can be hard to interact with

Question Answering over KB



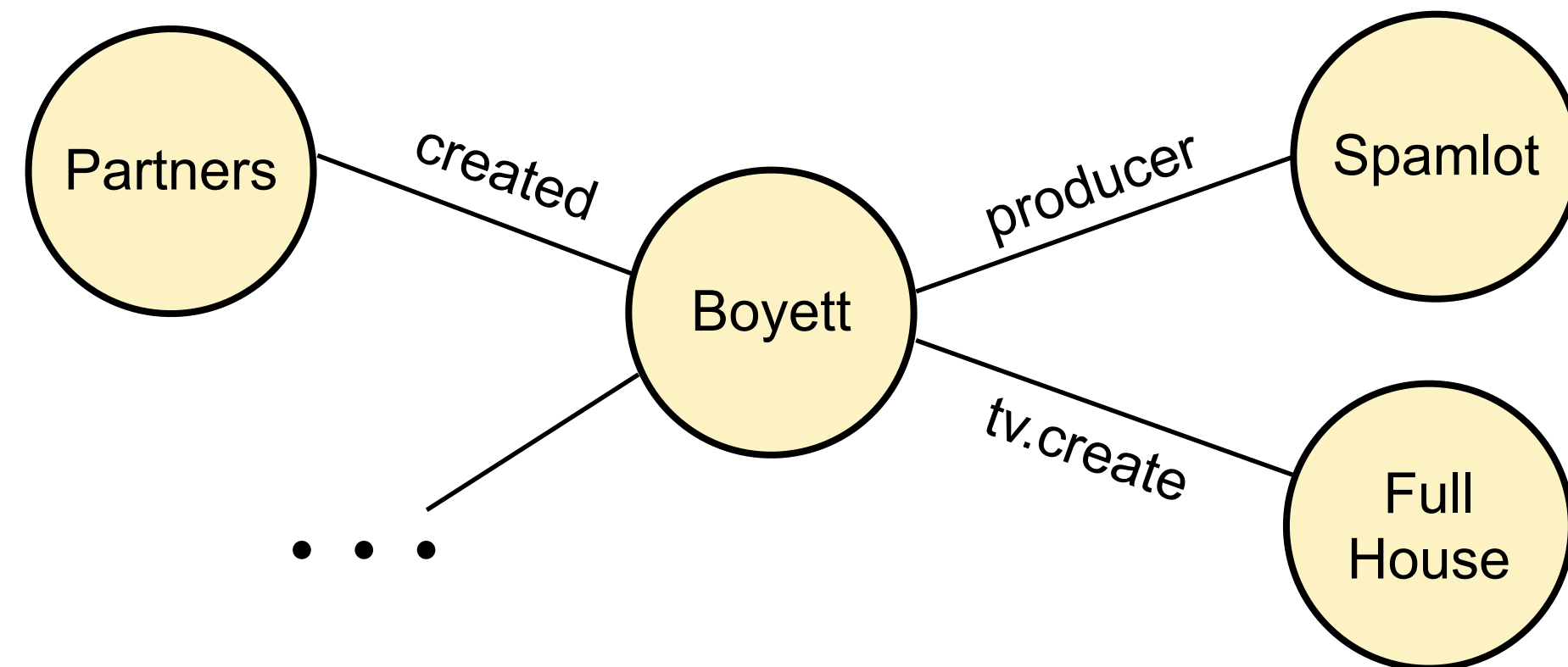
Q: *How many tv series has Boyett produced?*

LF: `Count (And theater_prod (Join create Boyett))`

- ▶ KBs are reliable sources of knowledge, but can be hard to interact with
- ▶ KBQA provides user-friendly access to KBs by converting a query to a logical form (LF) that can be executed over the KB

Question Answering over KB

Training



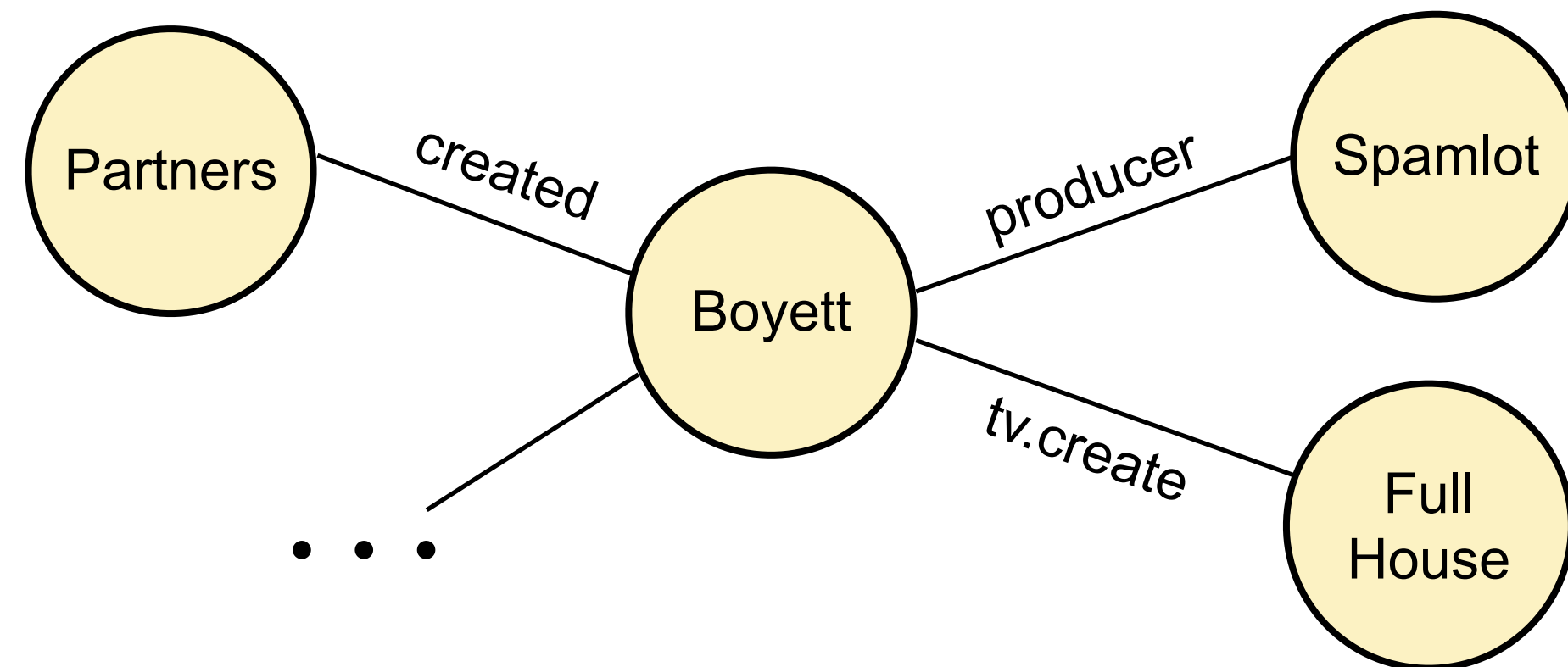
Q: *How many tv series has Boyett produced?*

LF: `Count (And theater_prod (Join create Boyett))`

- ▶ KBs are reliable sources of knowledge, but can be hard to interact with
- ▶ KBQA provides user-friendly access to KBs by converting a query to a logical form (LF) that can be executed over the KB
- ▶ Existing systems struggle in generalizing to new domains with **unseen KB schema items**

Question Answering over KB

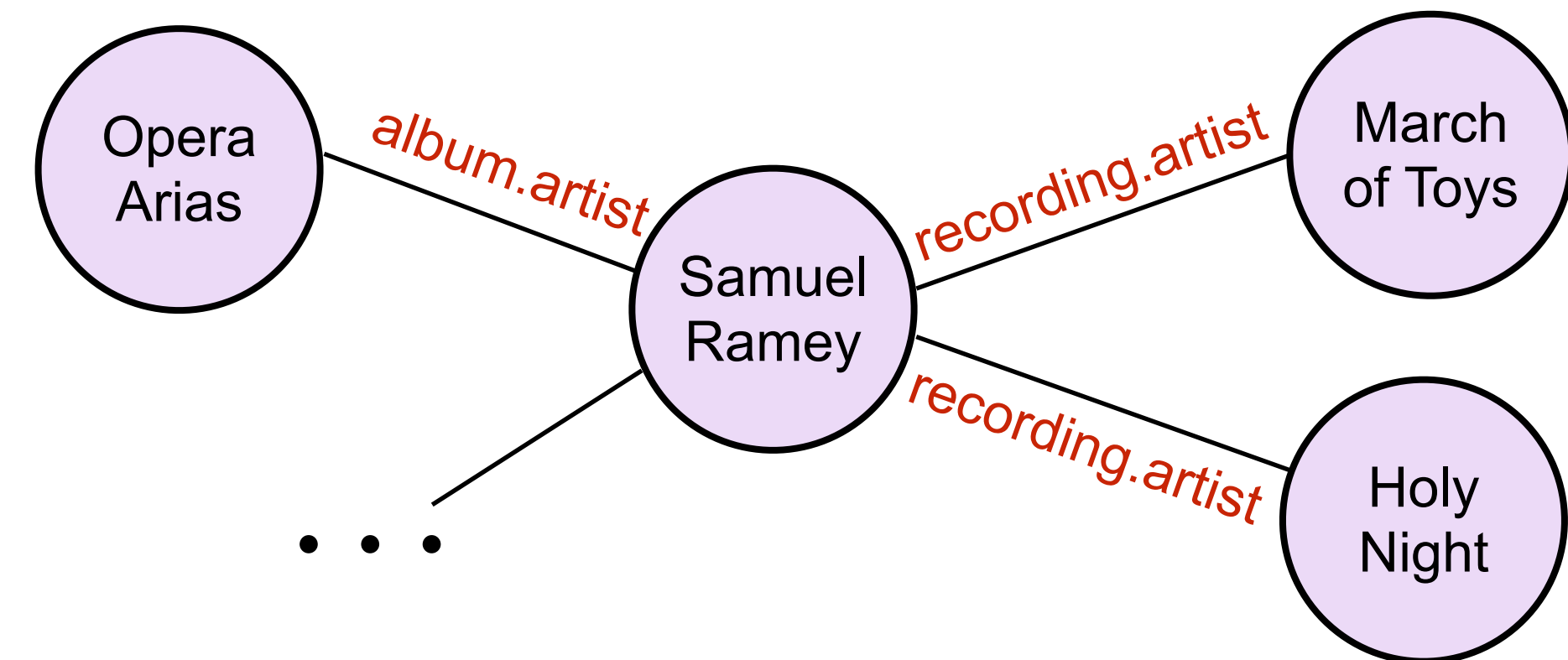
Training



Q: *How many tv series has Boyett produced?*

LF: `Count (And theater_prod (Join create Boyett))`

Zero-shot Generalization

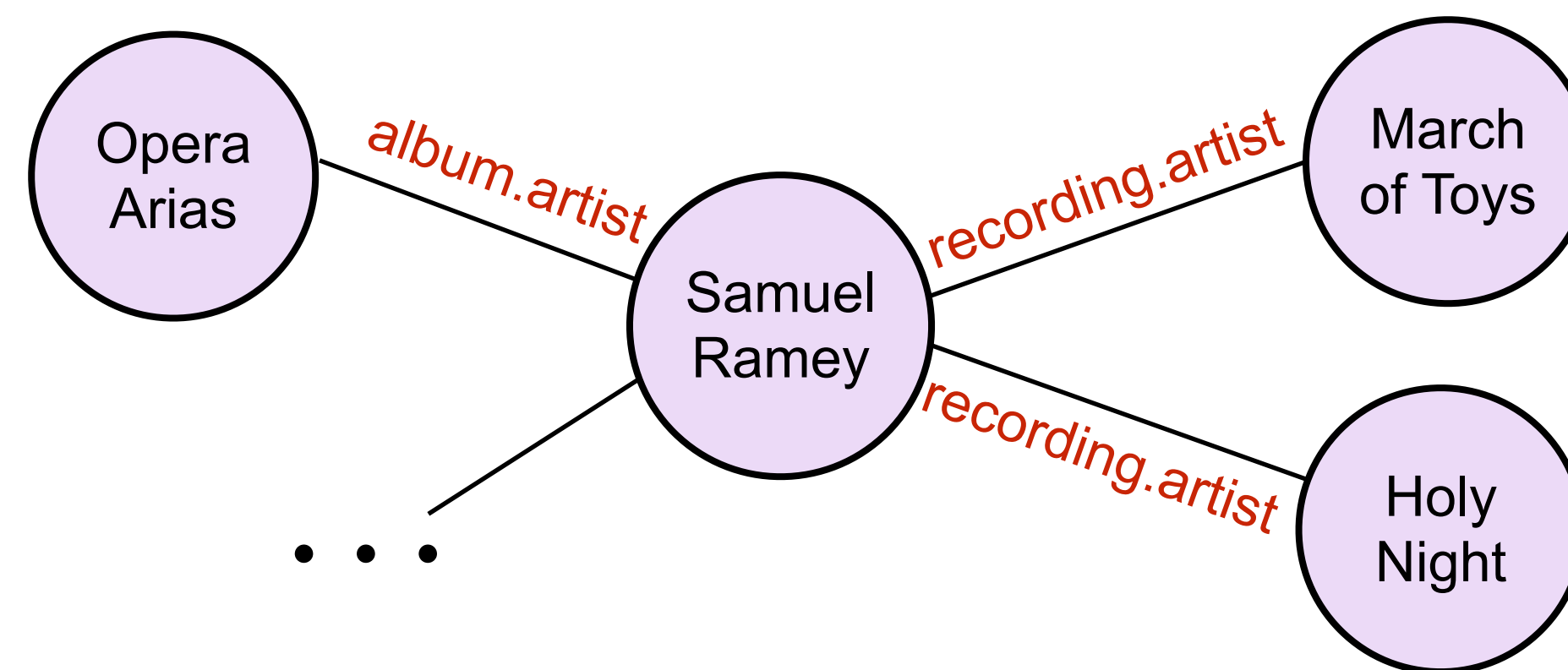


Q: *How many recordings has Ramey produce?*

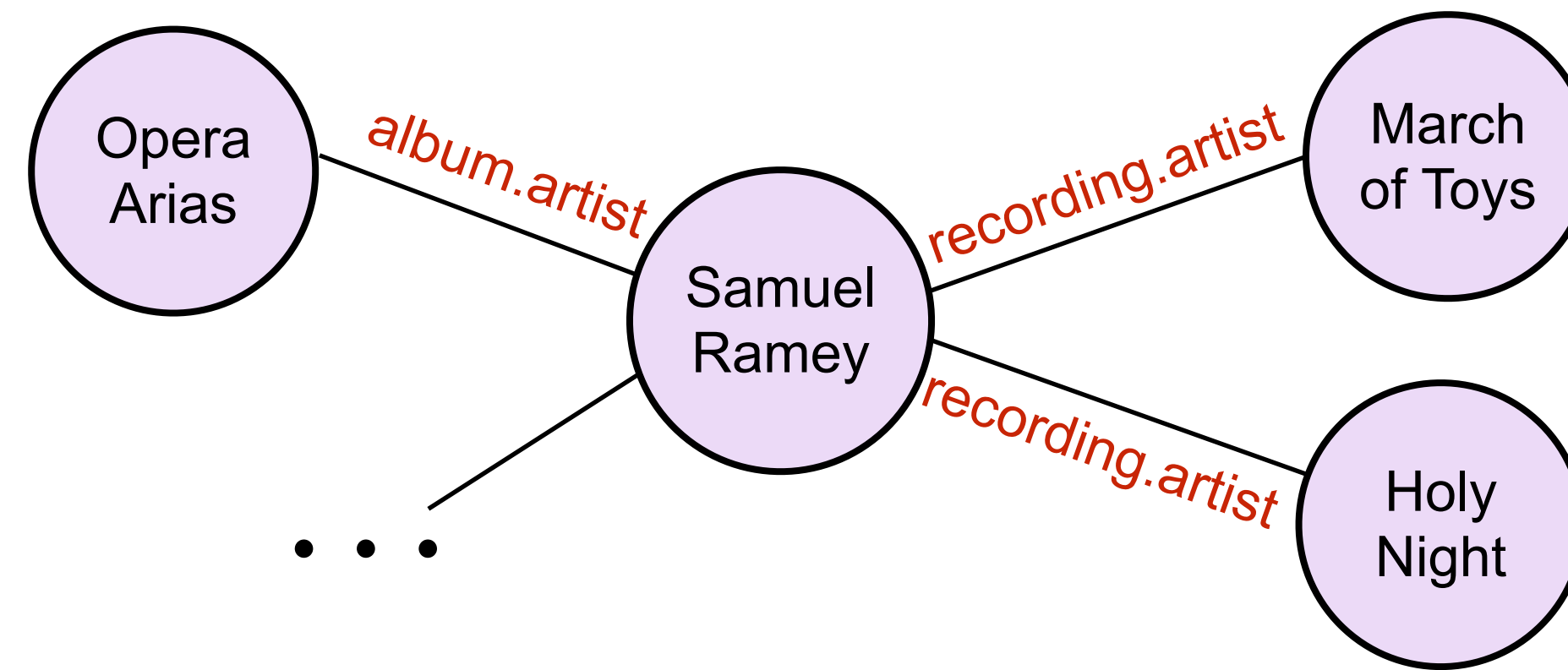
LF: `Count (And recording (Join recording.artist Ramey))`

- ▶ KBs are reliable sources of knowledge, but can be hard to interact with
- ▶ KBQA provides user-friendly access to KBs by converting a query to a logical form (LF) that can be executed over the KB
- ▶ Existing systems struggle in generalizing to new domains with **unseen KB schema items**

Generalization Challenge

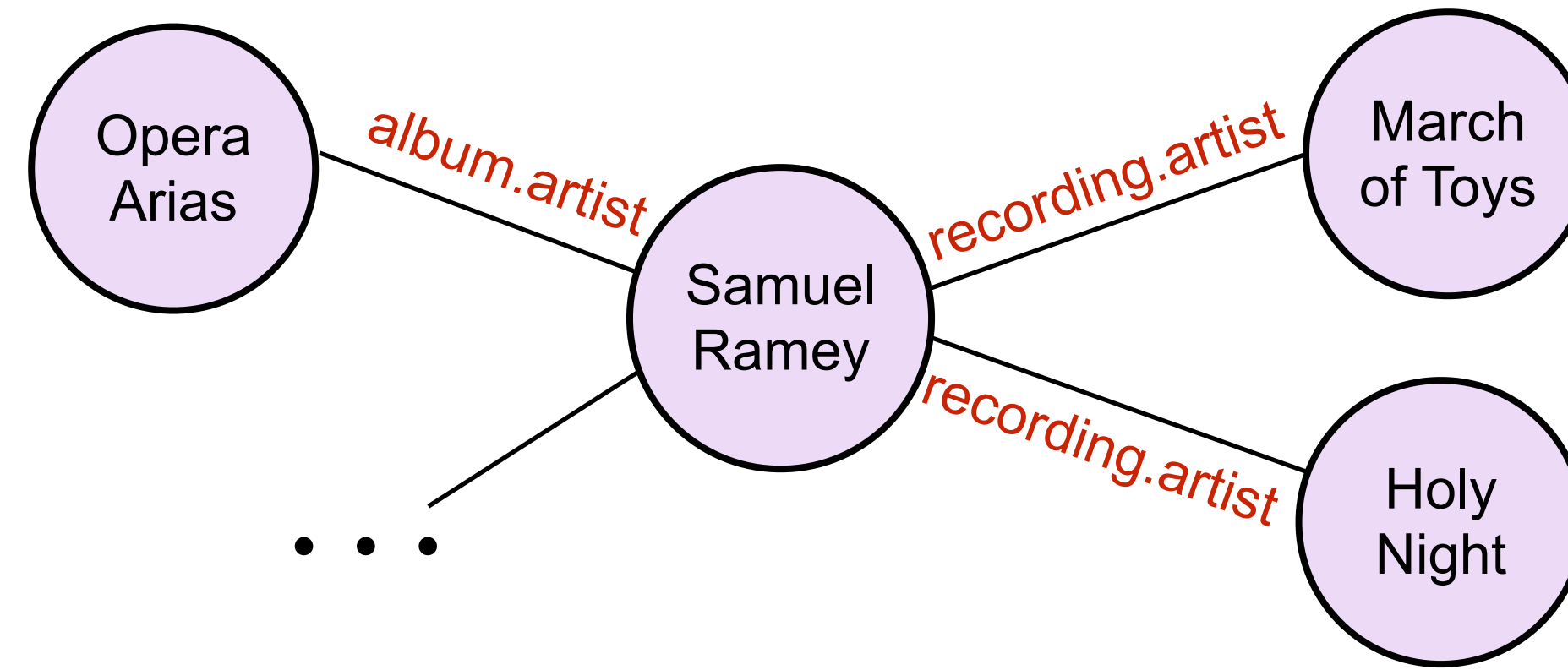


Generalization Challenge



- It is difficult for parsers (e.g., Seq-to-Seq models) to directly generate unseen schema items

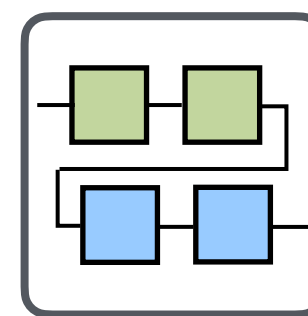
Generalization Challenge



Query

*How many recordings
has Ramey produce?*

Parser



Target LF

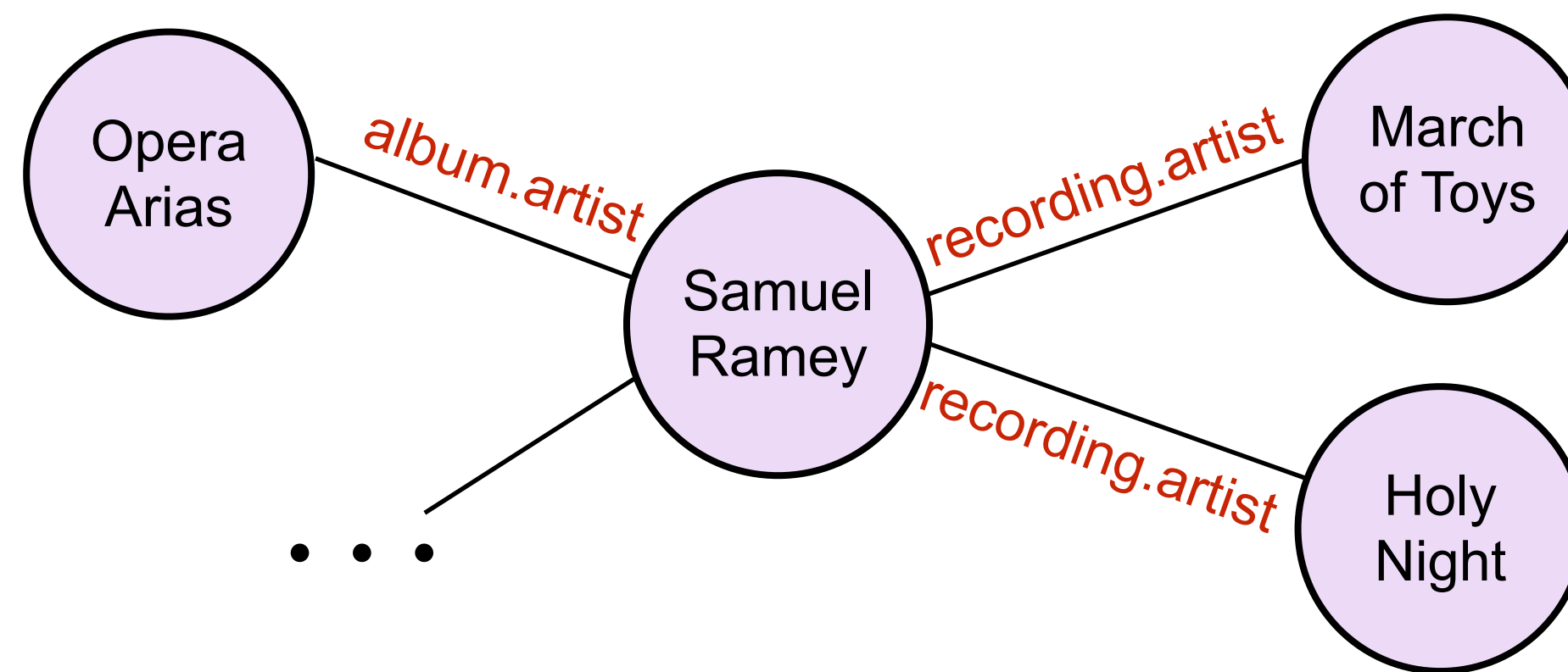
Count (And recording (Join
recording.artist Ramey))



- It is difficult for parsers (e.g., Seq-to-Seq models) to directly generate unseen schema items



Generalization Challenge (Cont'd)

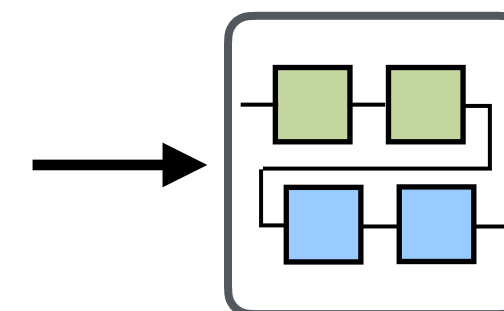


Query

*How many recordings
has Ramey produce?*



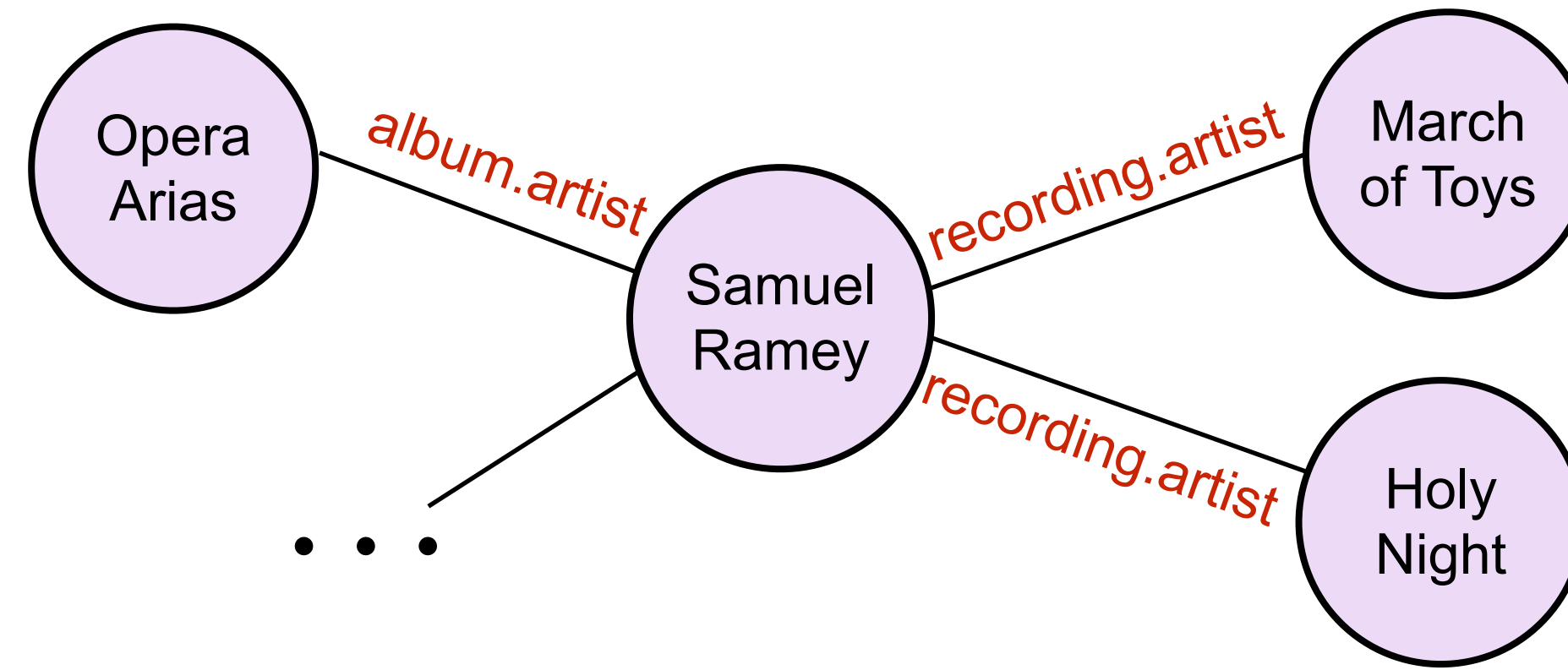
Parser



Target LF

`Count (And recording (Join
recording.artist Ramey))`

Generalization Challenge (Cont'd)



Query

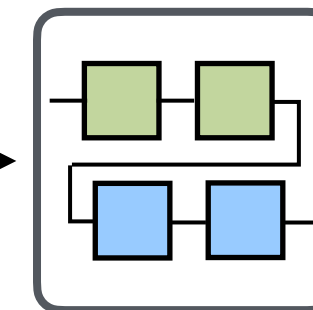
*How many recordings
has Ramey produce?*

Enumerated LFs

```

(Join recording.artist Ramey)
(Join album.artist Ramey)
...
  
```

Parser



Target LF

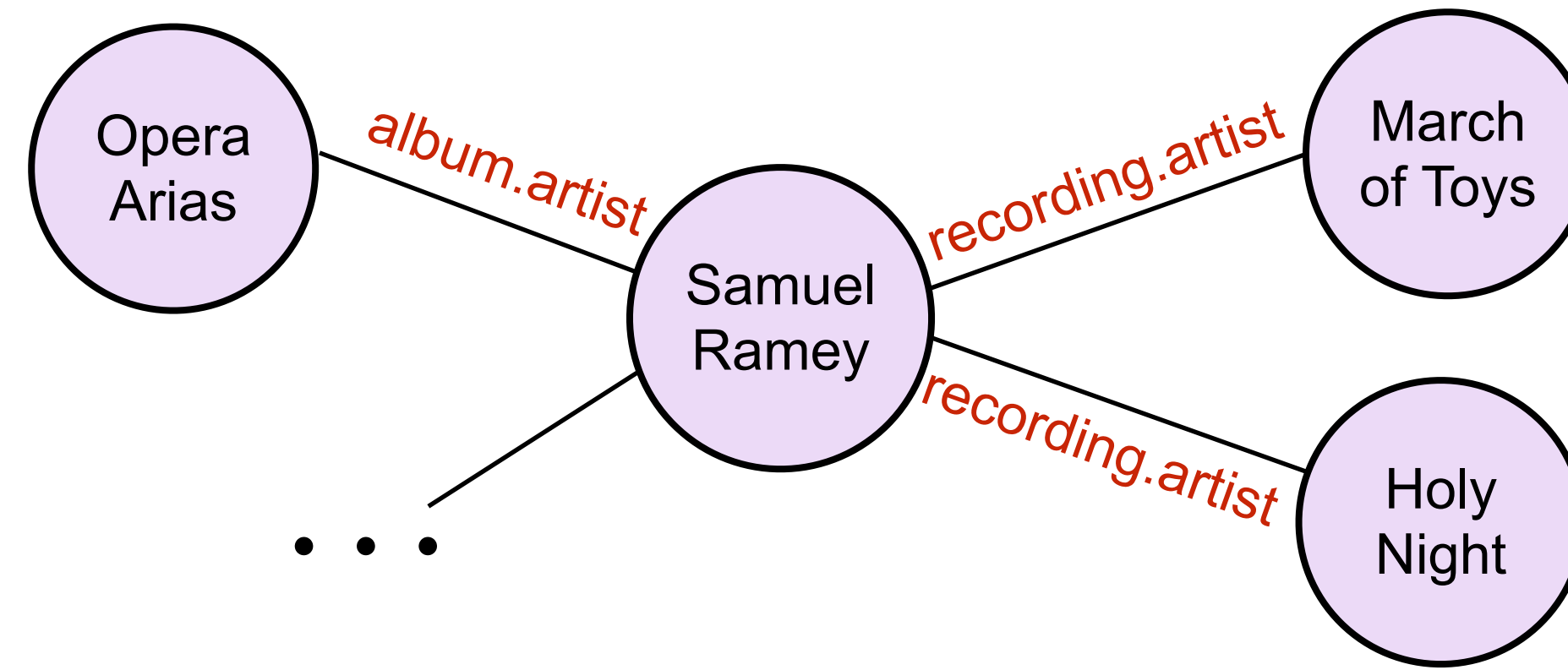
```

Count (And recording (Join
recording.artist Ramey))
  
```



Let the parser be conditioned on inputs that are closely related to the target LF

Generalization Challenge (Cont'd)



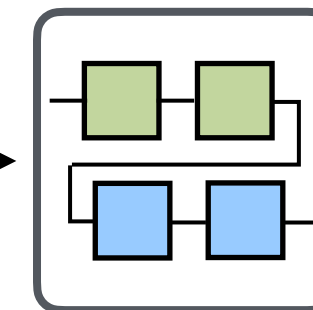
Query

*How many recordings
has Ramey produce?*

Enumerated LFs

(Join recording.artist Ramey)
(Join album.artist Ramey)
...

Parser



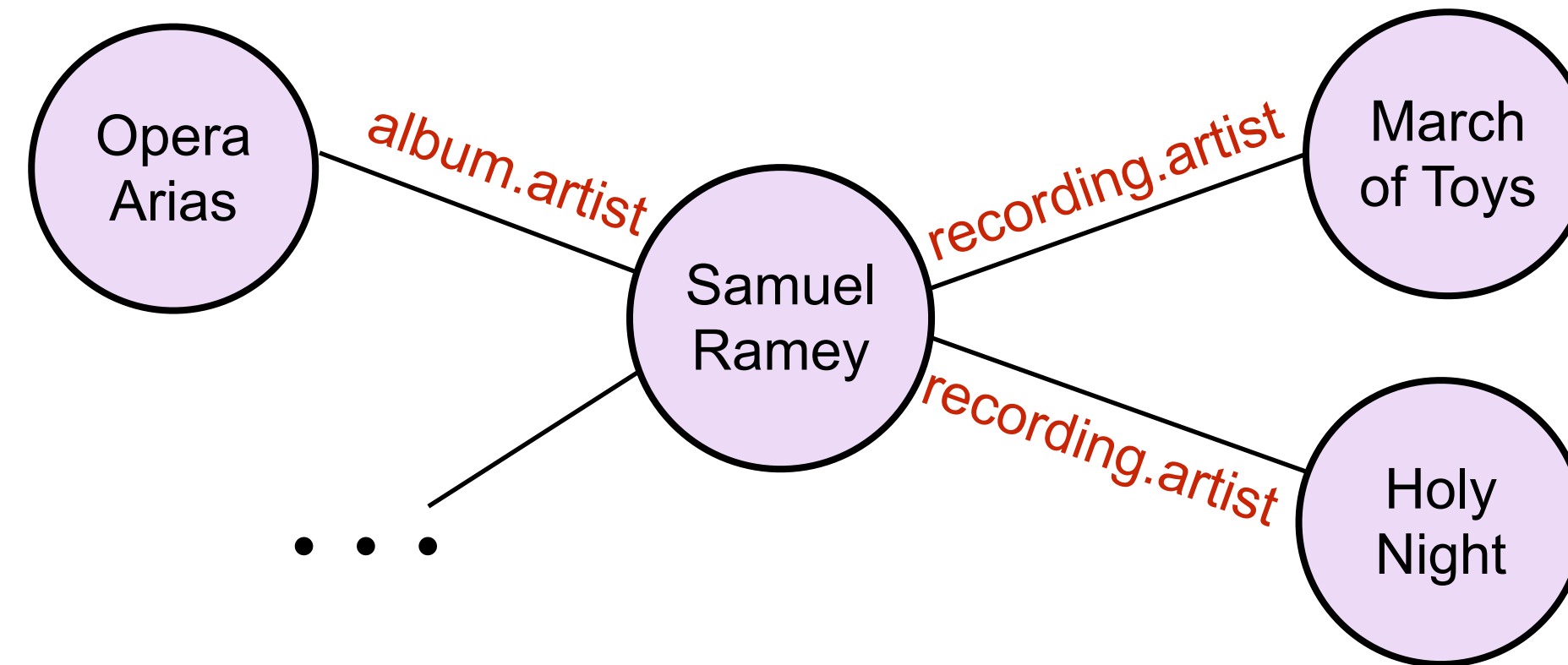
Target LF

Count (And recording (Join
recording.artist Ramey))



- ▶ Let the parser be conditioned on inputs that are closely related to the target LF
 - ▶ Enumerate some candidate LFs by querying the KB

Generalization Challenge (Cont'd)



Query

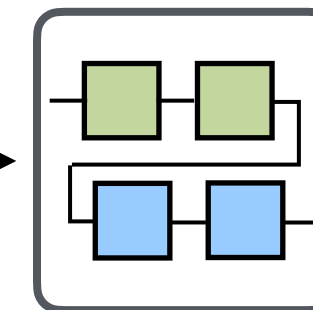
*How many recordings
has Ramey produce?*

Enumerated LFs

```

(Join recording.artist Ramey)
(Join album.artist Ramey)
...
  
```

Parser



Target LF

```

Count (And recording (Join
recording.artist Ramey))
  
```



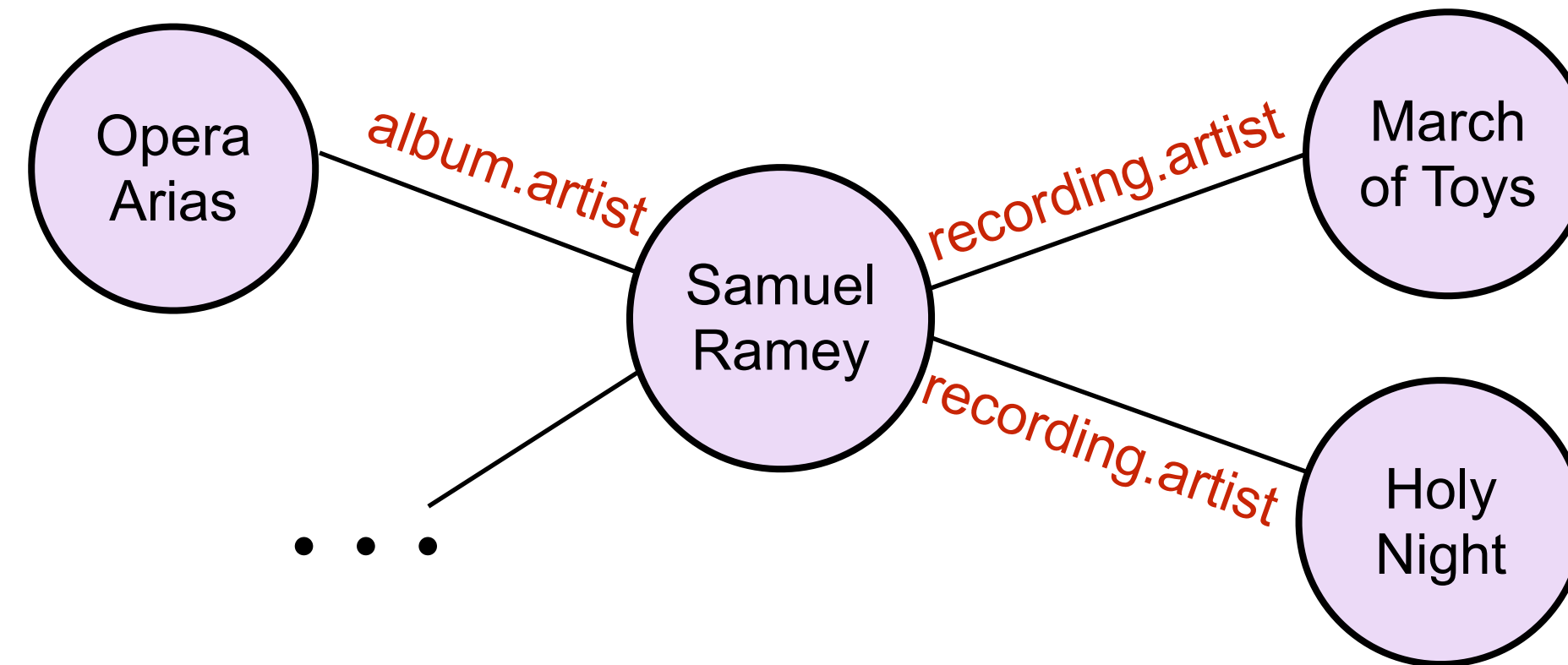
▶ Let the parser be conditioned on inputs that are closely related to the target LF

▶ Enumerate some candidate LFs by querying the KB



▶ Why not enumerate all possible LFs and select the target?

Generalization Challenge (Cont'd)



Query

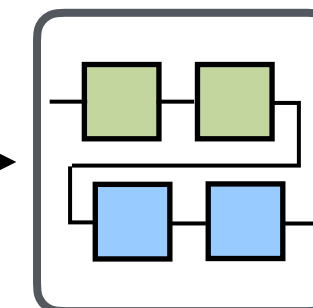
*How many recordings
has Ramey produce?*

Enumerated LFs

```

(Join recording.artist Ramey)
(Join album.artist Ramey)
...
  
```

Parser



Target LF

```

Count (And recording (Join
recording.artist Ramey))
  
```



▶ Let the parser be conditioned on inputs that are closely related to the target LF

▶ Enumerate some candidate LFs by querying the KB



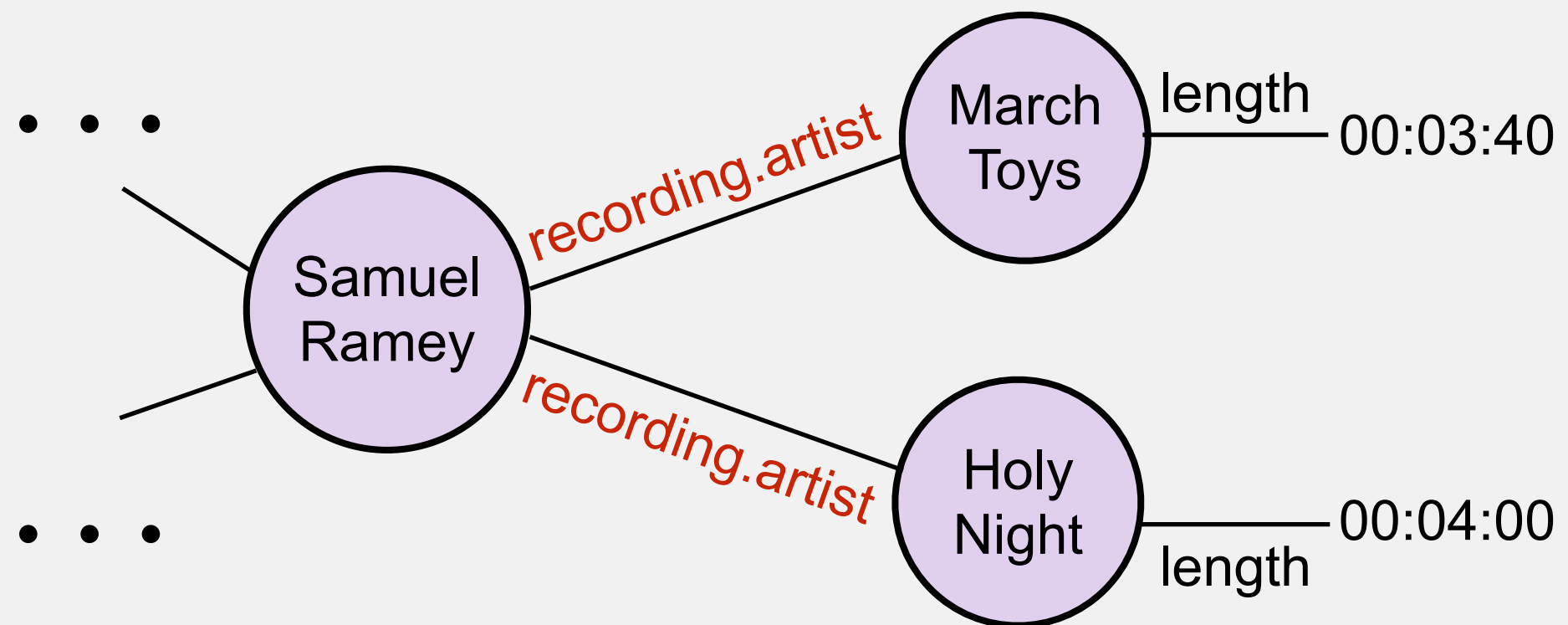
▶ Why not enumerate all possible LFs and select the target?

▶ Enumeration **cannot** guarantee to cover the target LF due to the **prohibitively large space**



Rank-and-Generate Framework

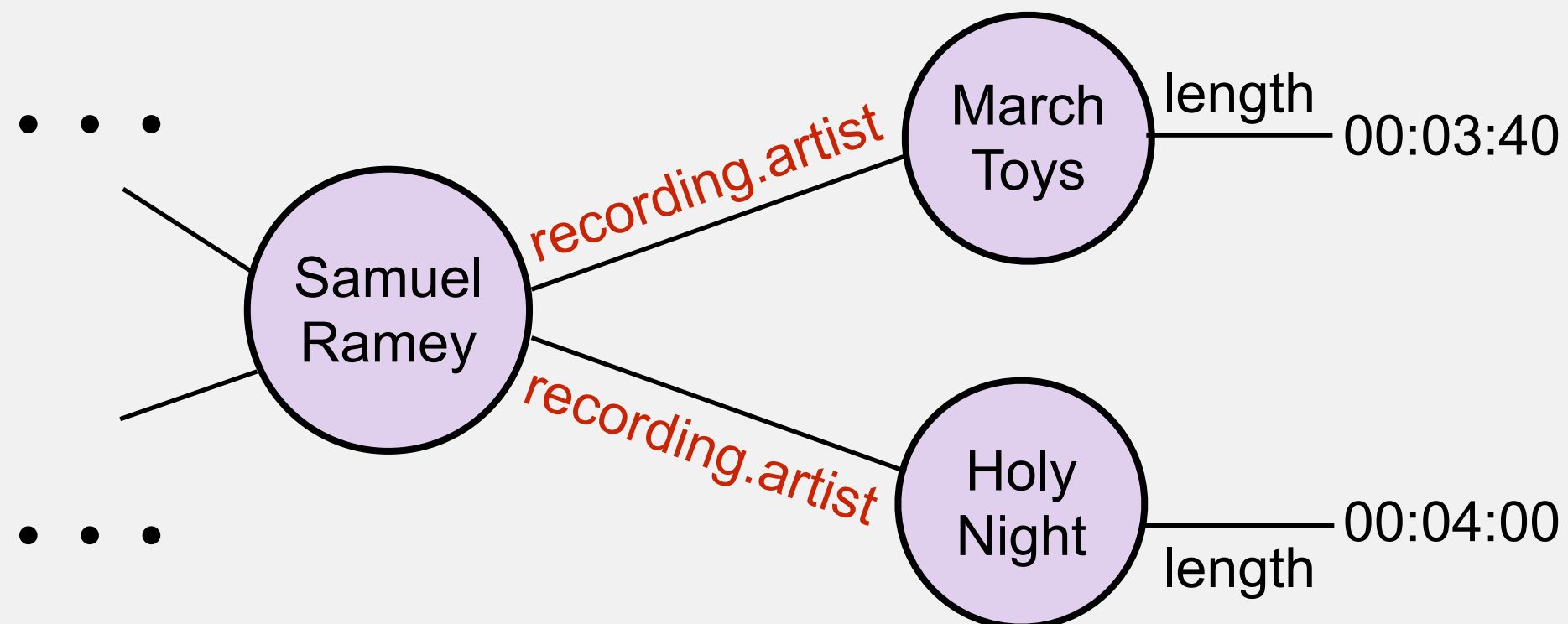
What is the shortest recording by Ramey?








Rank-and-Generate Framework

What is the shortest recording by Ramey?

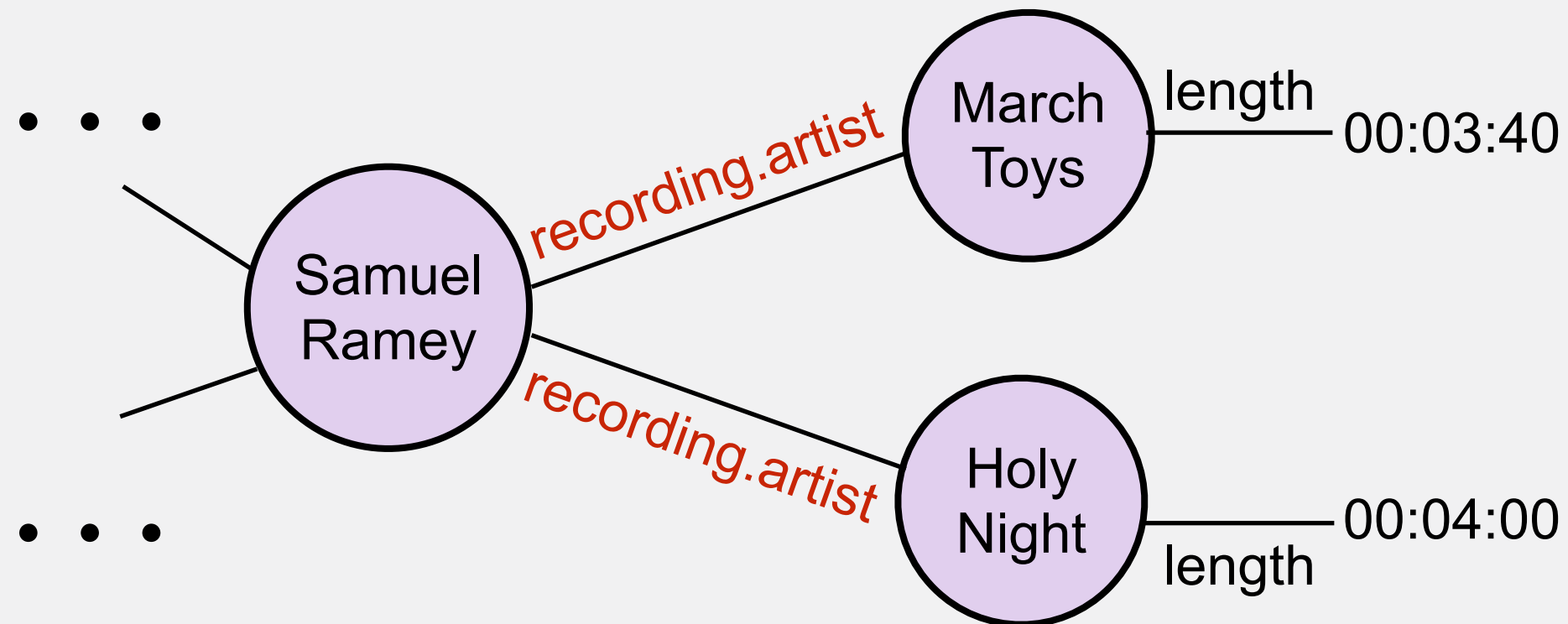


Enumerated Logical Forms

-  `(AND music.album (JOIN album.artist Ramey))`
-  `(JOIN (R recording.length) (JOIN recording.artist Ramey))`
-  `(AND music.recording (JOIN (R artist.track) Ramey))`
- `...`

Rank-and-Generate Framework

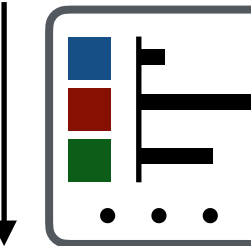
What is the shortest recording by Ramey?



Enumerated Logical Forms

- (AND music.album (JOIN album.artist Ramey))
- (JOIN (R recording.length) (JOIN recording.artist Ramey))
- (AND music.recording (JOIN (R artist.track) Ramey))
- ...

Ranker



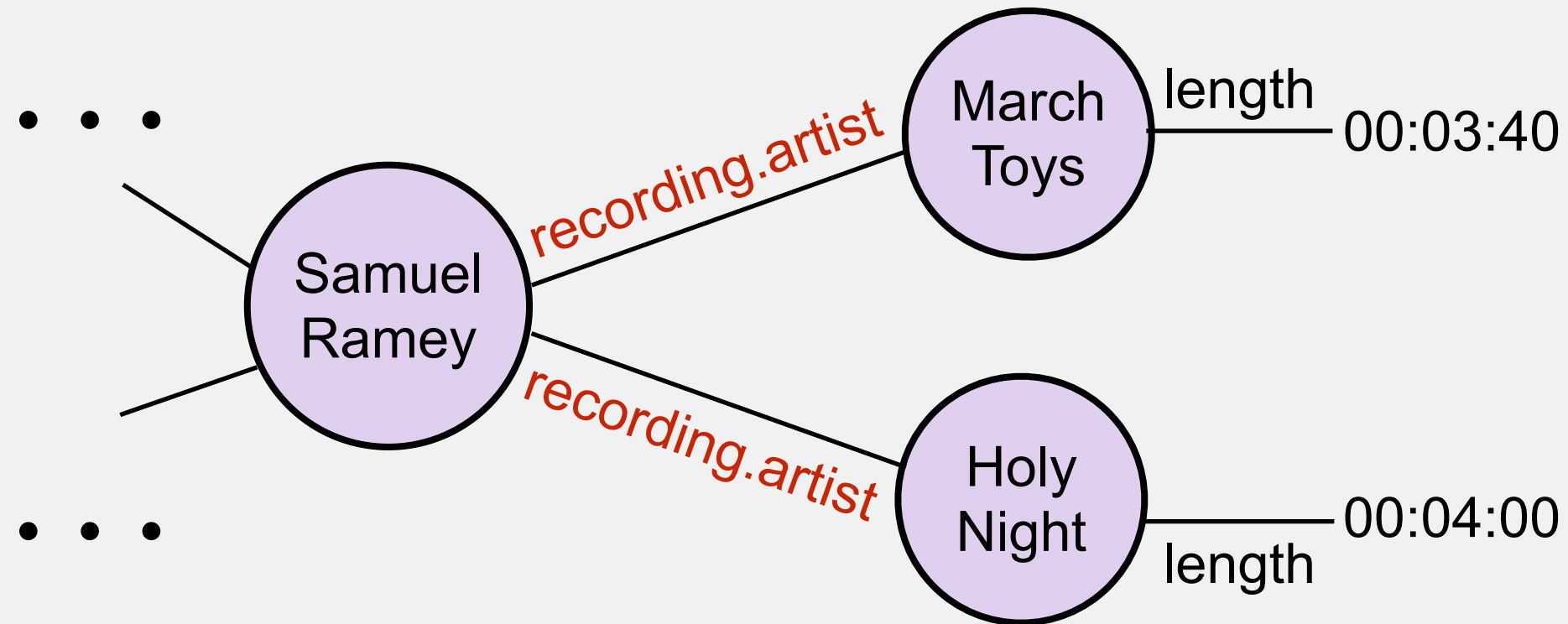
Top-Ranked Candidates

- (JOIN (R recording.length) (JOIN recording.artist Ramey))
- (AND music.recording (JOIN recording.artist Ramey))
- (AND music.album (JOIN album.artist Ramey))






Rank-and-Generate Framework

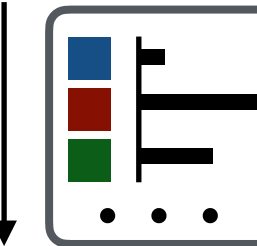
What is the shortest recording by Ramey?






Enumerated Logical Forms

-  (AND music.album (JOIN album.artist Ramey))
-  (JOIN (R recording.length) (JOIN recording.artist Ramey))
-  (AND music.recording (JOIN (R artist.track) Ramey))
- ...

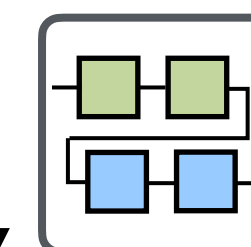
Ranker



Top-Ranked Candidates

-  (JOIN (R recording.length) (JOIN recording.artist Ramey))
-  (AND music.recording (JOIN recording.artist Ramey))
-  (AND music.album (JOIN album.artist Ramey))

Generator

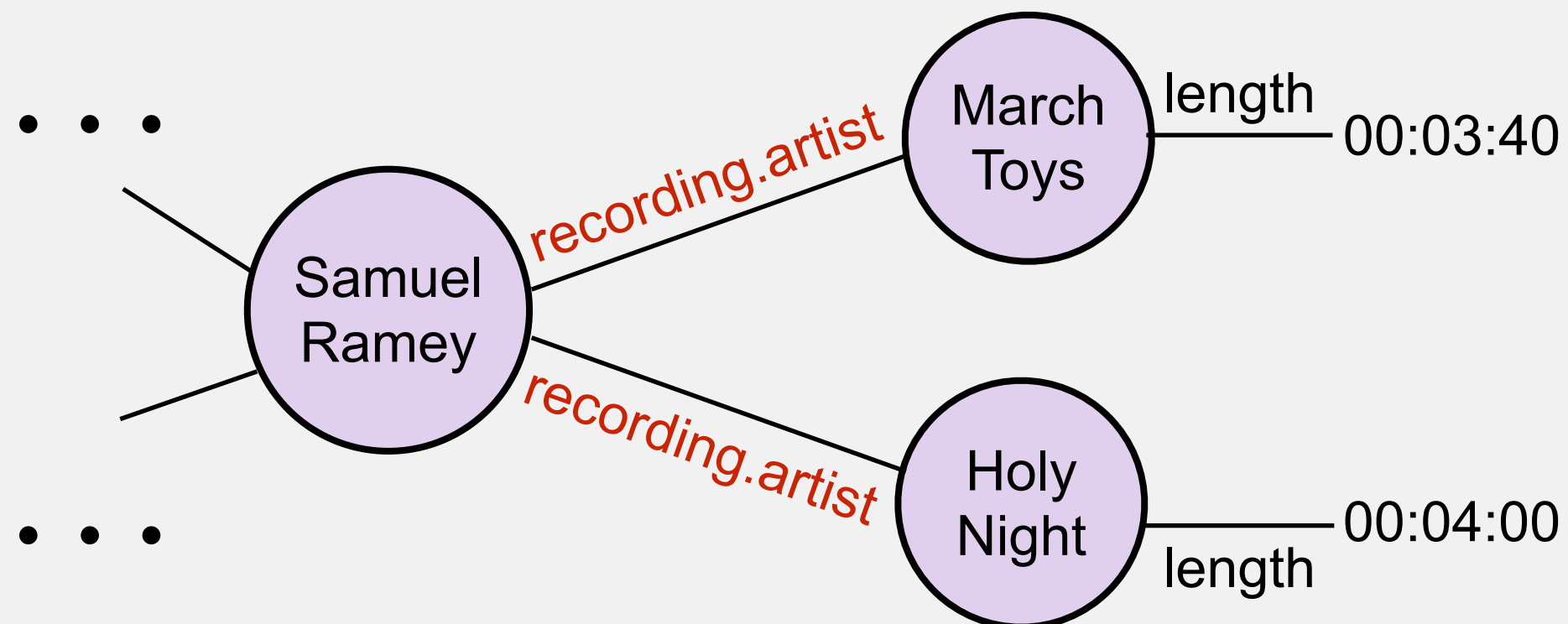


Target Logical Form

```
(ARGMIN (AND music.recording
              (JOIN recording.artist Ramey)) recording.length)
```

Rank-and-Generate Framework

What is the shortest recording by Ramey?

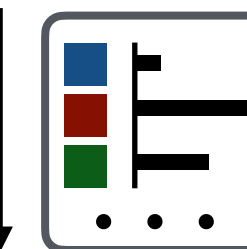


- **Ranker:** select closely related LF candidates for the generator to be conditioned on

Enumerated Logical Forms

- (AND music.album (JOIN album.artist Ramey))
- (JOIN (R recording.length) (JOIN recording.artist Ramey))
- (AND music.recording (JOIN (R artist.track) Ramey))
- ...

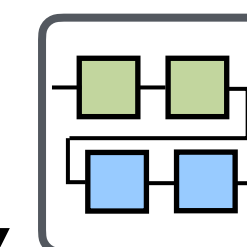
Ranker



Top-Ranked Candidates

- (JOIN (R recording.length) (JOIN recording.artist Ramey))
- (AND music.recording (JOIN recording.artist Ramey))
- (AND music.album (JOIN album.artist Ramey))

Generator

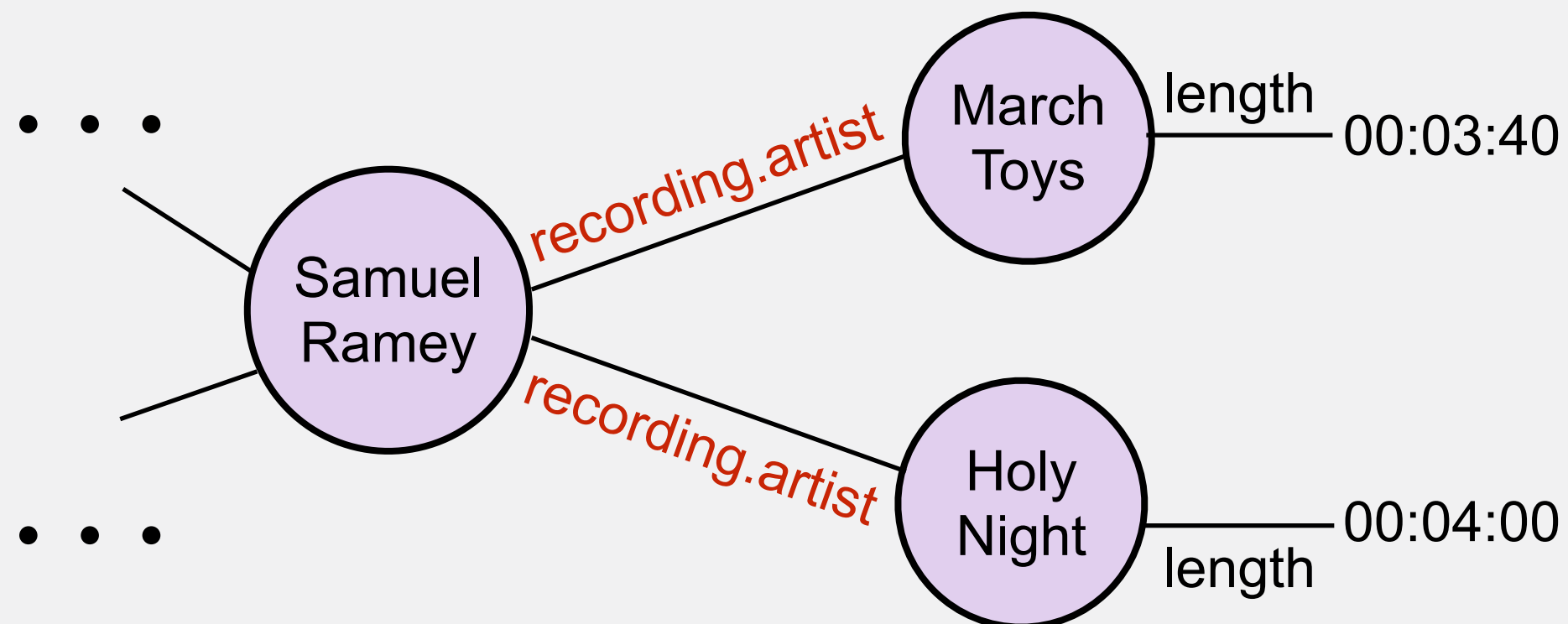


Target Logical Form

(ARGMIN (AND music.recording
(JOIN recording.artist Ramey)) recording.length)

Rank-and-Generate Framework

What is the shortest recording by Ramey?

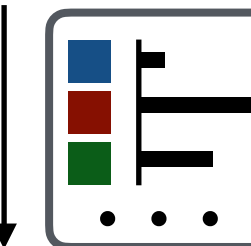


- ▶ **Ranker:** select closely related LF candidates for the generator to be conditioned on
- ▶ **Generator:** refine the top-ranked candidates by fixing uncovered compositions or operators

Enumerated Logical Forms

- (AND music.album (JOIN album.artist Ramey))
- (JOIN (R recording.length) (JOIN recording.artist Ramey))
- (AND music.recording (JOIN (R artist.track) Ramey))
- ...

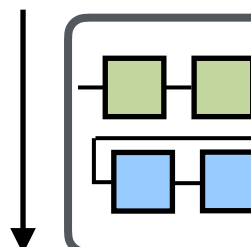
Ranker



Top-Ranked Candidates

- (JOIN (R recording.length) (JOIN recording.artist Ramey))
- (AND music.recording (JOIN recording.artist Ramey))
- (AND music.album (JOIN album.artist Ramey))

Generator

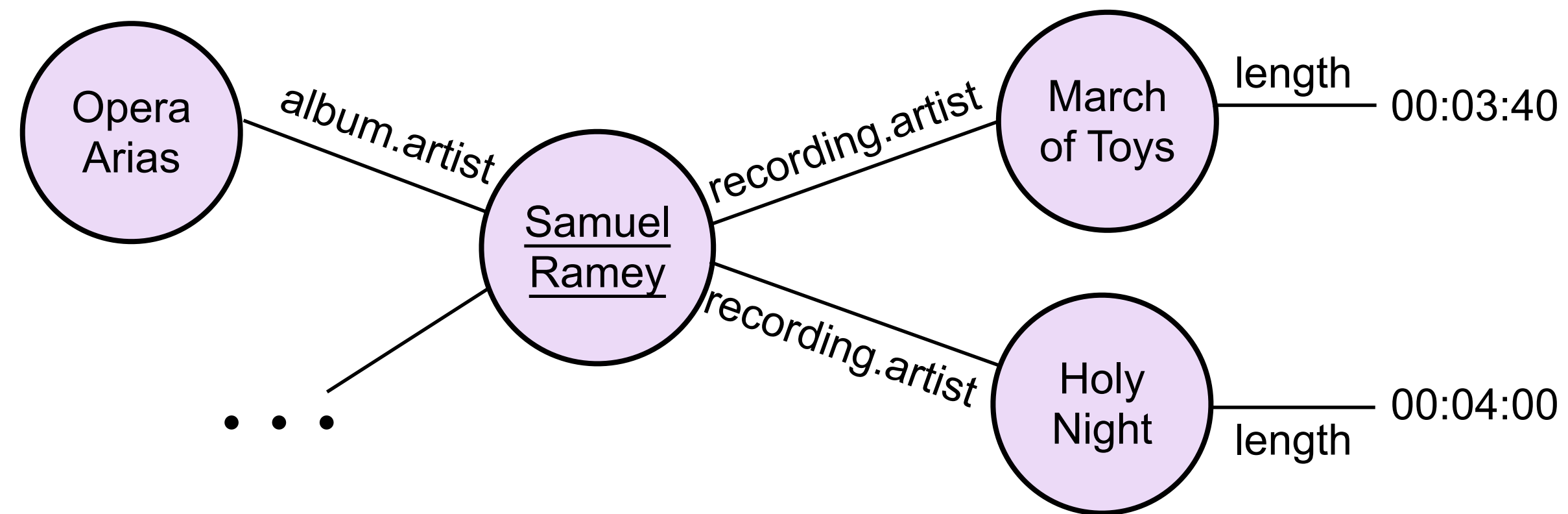


Target Logical Form

(ARGMIN (AND music.recording
(JOIN recording.artist Ramey)) recording.length)

Enumeration

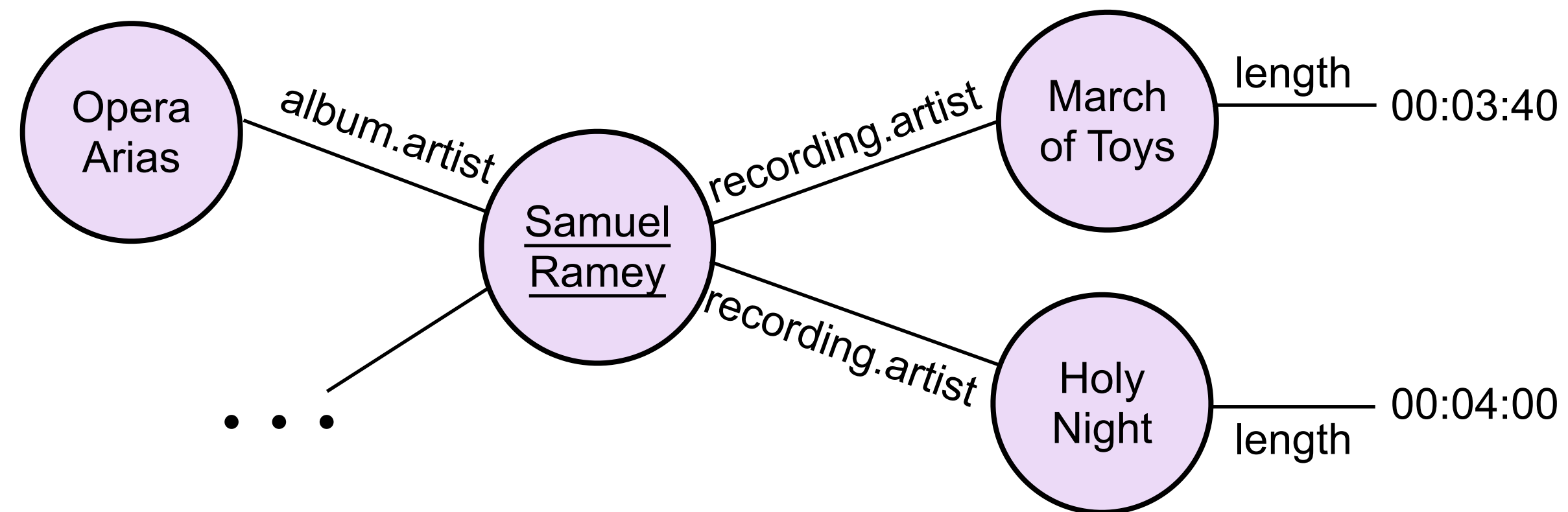
What is the shortest recording by Ramey?



Enumeration

- ▶ Enumerate paths reachable within two hops starting from the topic entity

What is the shortest recording by Ramey?



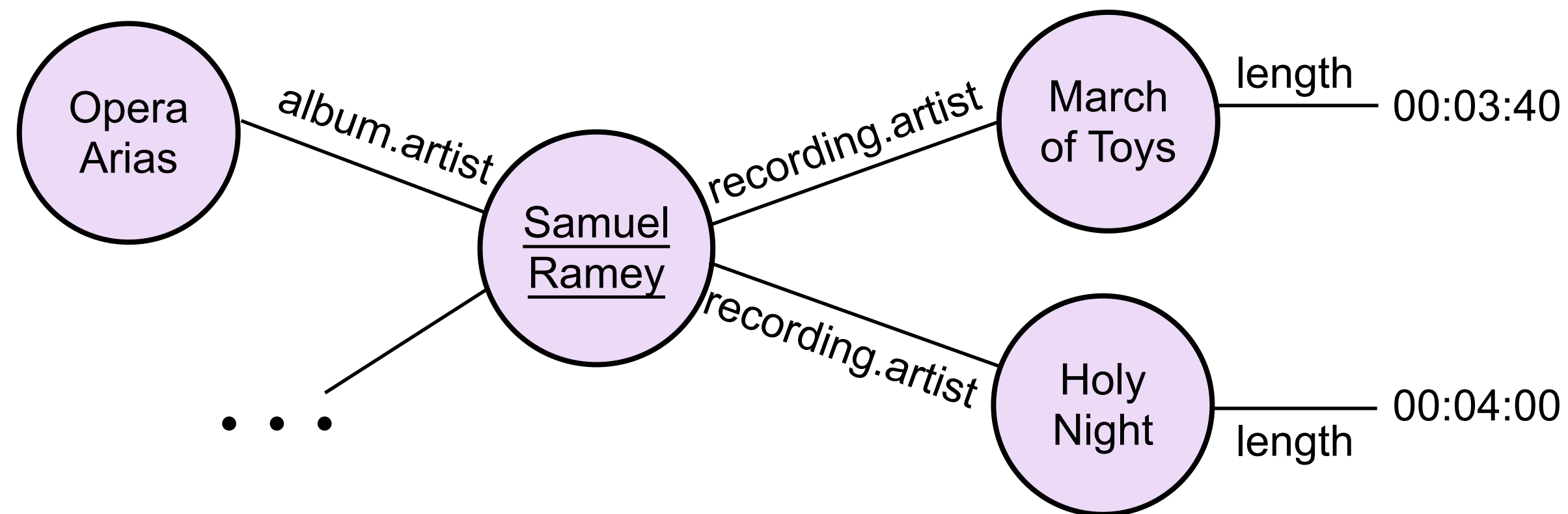
One-Hop `(Join recording.artist Ramey)`

Two-Hop `(And (Join length 00:03:40) (Join recording.artist Ramey))`

Enumeration

- ▶ Enumerate paths reachable within two hops starting from the topic entity
- ▶ Only covering part of the possible compositions (e.g., argmin/max operation are excluded)

What is the shortest recording by Ramey?



One-Hop `(Join recording.artist Ramey)`

Two-Hop `(And (Join length 00:03:40) (Join recording.artist Ramey))`

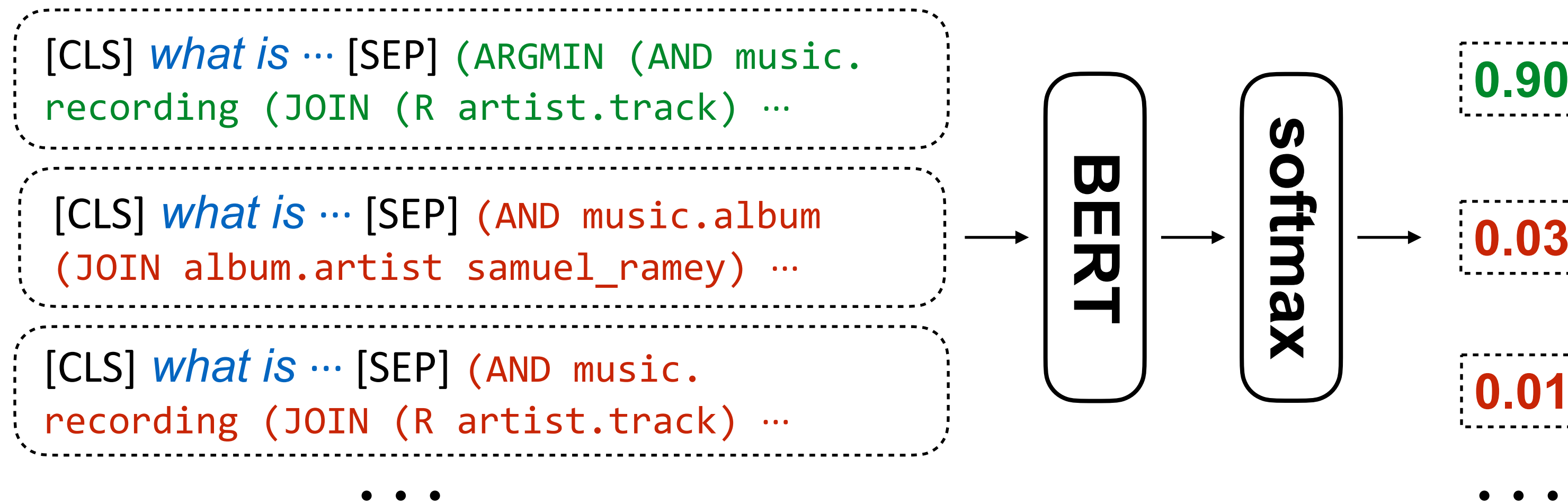


Contrastive Ranker

- ▶ Bert-based model scoring the concatenation of the question and a logical form

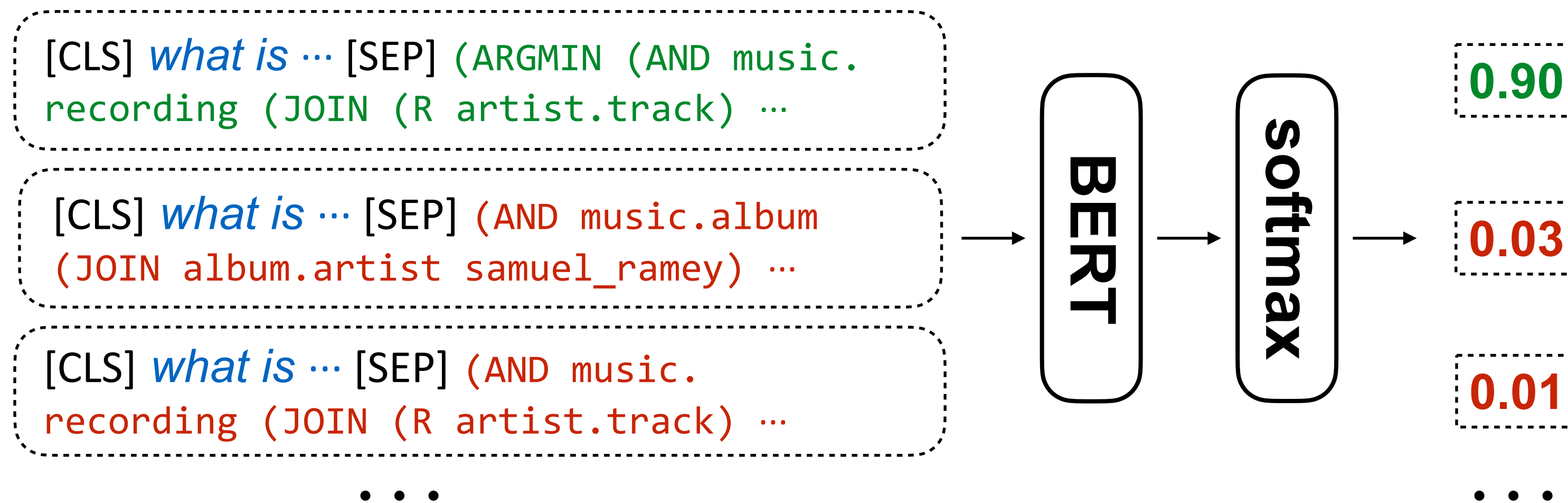
Contrastive Ranker

- Bert-based model scoring the concatenation of the question and a logical form



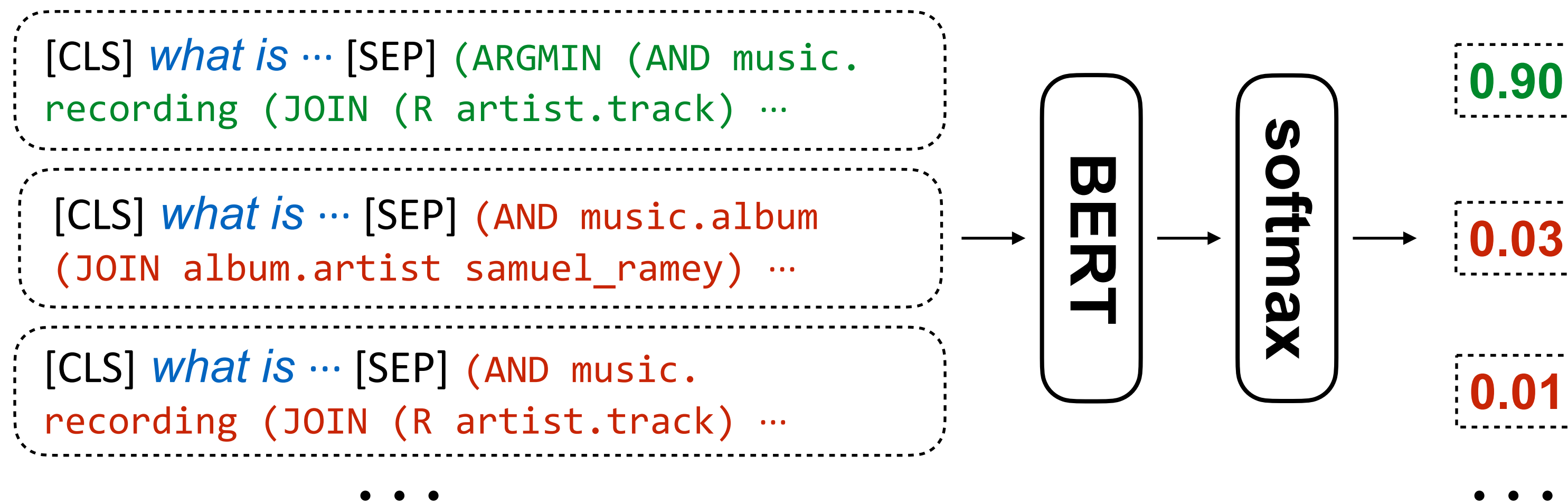
Contrastive Ranker

- ▶ Bert-based model scoring the concatenation of the question and a logical form
- ▶ Learns to contrast **ground truth** LF with **negative** LFs



Contrastive Ranker

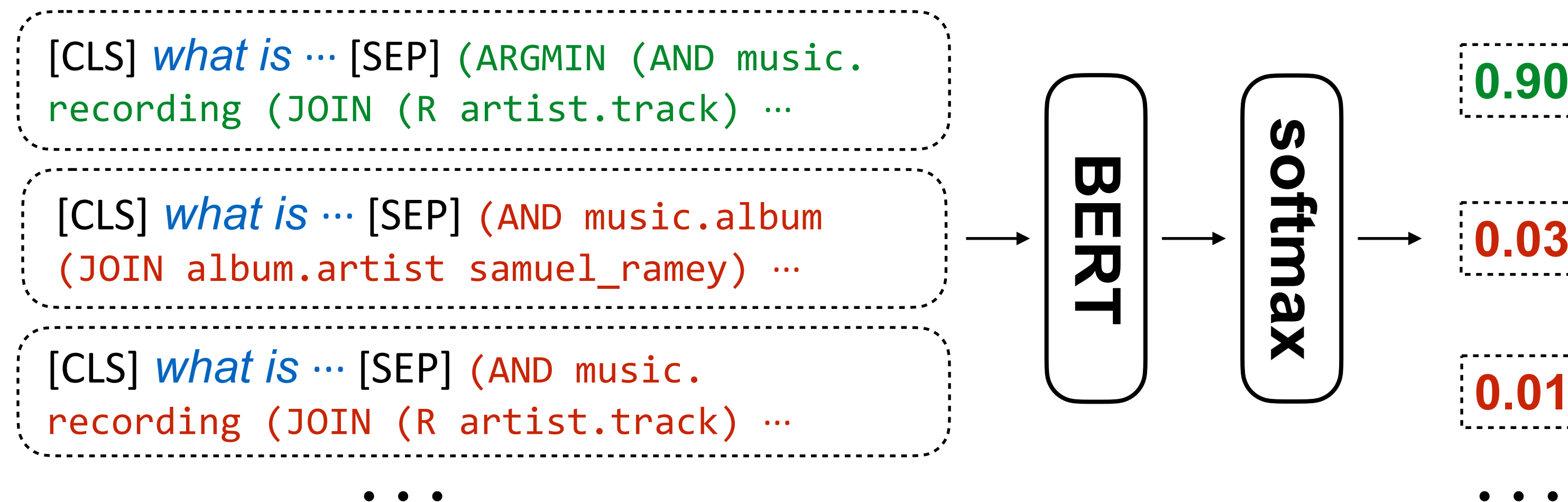
- ▶ Bert-based model scoring the concatenation of the question and a logical form
- ▶ Learns to contrast **ground truth** LF with **negative** LFs



- ▶ Sample negative LFs in each batch during training due to limited GPU memory

Contrastive Ranker

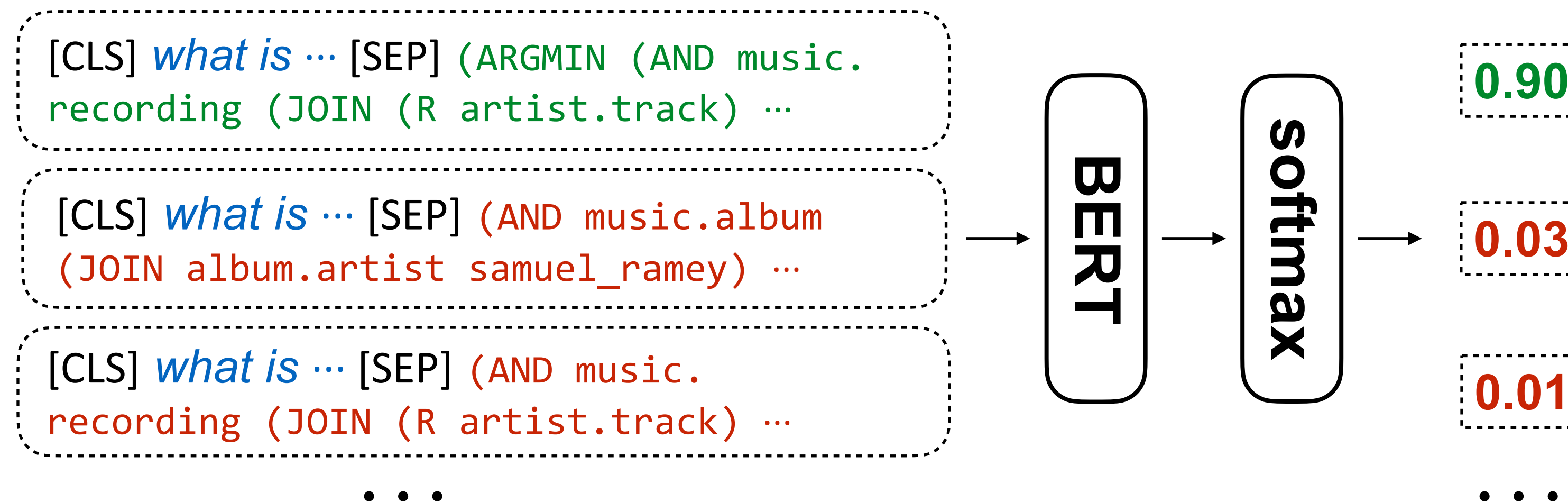
- ▶ Bert-based model scoring the concatenation of the question and a logical form
- ▶ Learns to contrast **ground truth** LF with **negative** LFs



- ▶ Sample negative LFs in each batch during training due to limited GPU memory
- ▶ **Random:** randomly sample

Contrastive Ranker

- ▶ Bert-based model scoring the concatenation of the question and a logical form
- ▶ Learns to contrast **ground truth** LF with **negative** LFs



- ▶ Sample negative LFs in each batch during training due to limited GPU memory
 - ▶ **Random:** randomly sample
 - ▶ **Bootstrapping:** First, warm-start the ranker with random negative samples. Next, select negative samples that are highly scored by the ranker



Generator

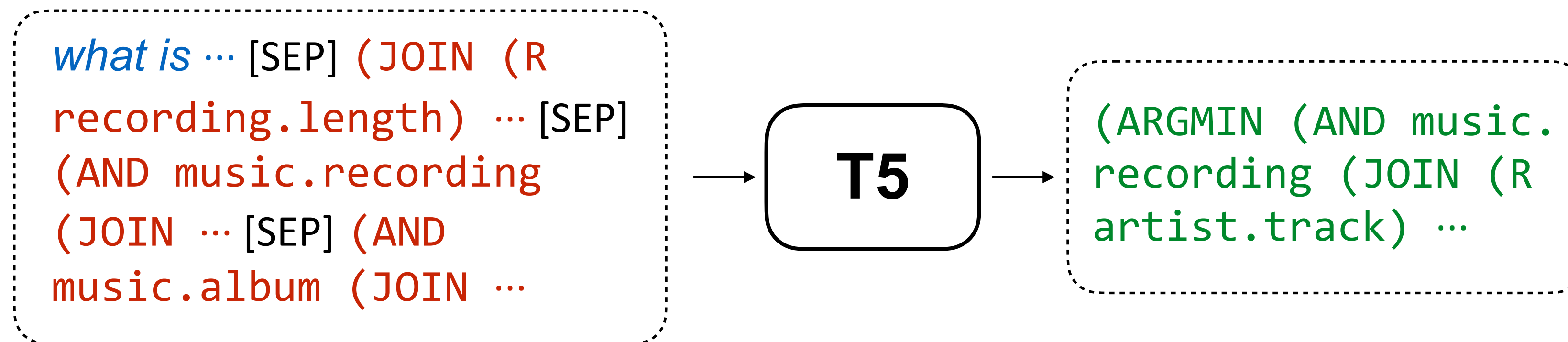


Generator

- ▶ T5 generation model conditioned on the concatenation of the question and top-ranked LFs

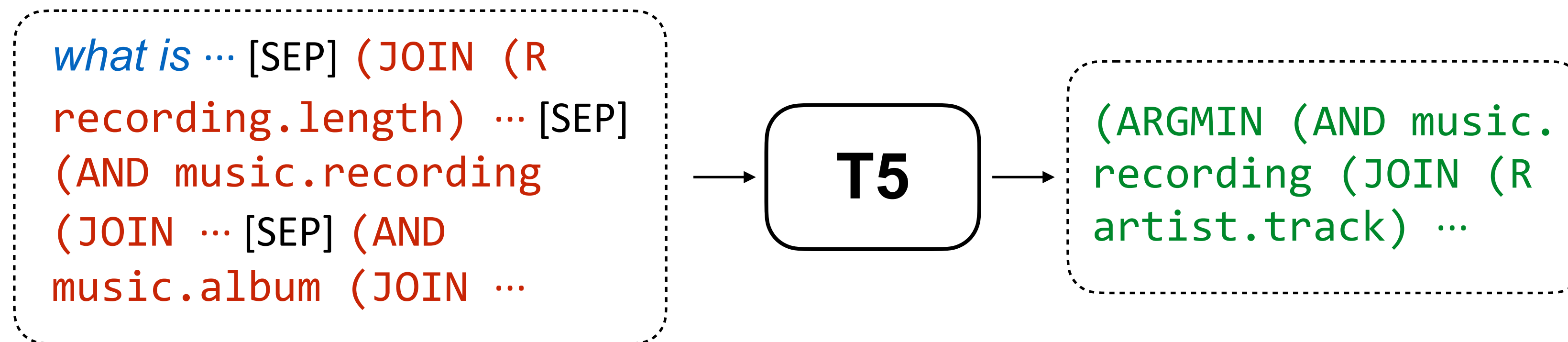
Generator

- ▶ T5 generation model conditioned on the concatenation of the question and top-ranked LFs



Generator

- ▶ T5 generation model conditioned on the concatenation of the question and top-ranked LFs
- ▶ Run the trained ranker over the training set to construct training data for the generator





Experiments



Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**



Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
 - ▶ i.i.d.

Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
 - ▶ i.i.d.
 - ▶ compositional: seen KB schema items, **unseen composition**

Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
- ▶ i.i.d.
- ▶ compositional: seen KB schema items, **unseen composition**

Training Data

what are the music recordings by Samuel Ramey?

```
(AND music.recording (JOIN recording.artist samuel_ramey))
```

what are the albums by Samuel Ramey?

```
(AND music.album (JOIN album.artist samuel_ramey))
```

Compositional Generalization

what are the albums by the artist who makes the recoding Holy Night?

```
(AND music.album (JOIN album.artist  
                    (JOIN (R recording.artist) holy_night)))
```

Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
 - ▶ i.i.d.
 - ▶ compositional: seen KB schema items, **unseen composition**
 - ▶ zero-shot generalization: **unseen KB schema items**

Training Data

what are the music recordings by Samuel Ramey?

```
(AND music.recording (JOIN recording.artist samuel_ramey))
```

what are the albums by Samuel Ramey?

```
(AND music.album (JOIN album.artist samuel_ramey))
```

Compositional Generalization

what are the albums by the artist who makes the recoding Holy Night?

```
(AND music.album (JOIN album.artist  
                    (JOIN (R recording.artist) holy_night)))
```

Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
- ▶ i.i.d.
- ▶ compositional: seen KB schema items, **unseen composition**
- ▶ zero-shot generalization: **unseen KB schema items**

Training Data

what are the music recordings by Samuel Ramey?

```
(AND music.recording (JOIN recording.artist samuel_ramey))
```

what are the albums by Samuel Ramey?

```
(AND music.album (JOIN album.artist samuel_ramey))
```

Compositional Generalization

what are the albums by the artist who makes the recoding Holy Night?

```
(AND music.album (JOIN album.artist  
                    (JOIN (R recording.artist) holy_night)))
```

Zero-Shot Generalization

what songs for tv did Samuel Ramey write lyrics for?

```
(AND tv.tv_song (JOIN composition.lyricist samuel_ramey))
```


Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
 - ▶ i.i.d.
 - ▶ compositional: seen KB schema items, **unseen composition**
 - ▶ zero-shot generalization: **unseen KB schema items**
- ▶ **WebQSP** (Yih et al., 2016): classical dataset in i.i.d. setting

Training Data

what are the music recordings by Samuel Ramey?

(AND **music.recording** (JOIN **recording.artist** samuel_ramey))

what are the albums by Samuel Ramey?

(AND **music.album** (JOIN **album.artist** samuel_ramey))

Compositional Generalization

what are the albums by the artist who makes the recoding Holy Night?

(AND **music.album** (JOIN **album.artist**
(JOIN (R **recording.artist**) holy_night)))

Zero-Shot Generalization

what songs for tv did Samuel Ramey write lyrics for?

(AND **tv.tv_song** (JOIN **composition.lyricist** samuel_ramey))

Experiments

- ▶ **GrailQA** (Gu et al., 2021): A new dataset focused on **generalization**
 - ▶ i.i.d.
 - ▶ compositional: seen KB schema items, **unseen composition**
 - ▶ zero-shot generalization: **unseen KB schema items**
- ▶ **WebQSP** (Yih et al., 2016): classical dataset in i.i.d. setting
- ▶ **Metric:** F1 score

Training Data

what are the music recordings by Samuel Ramey?

(AND **music.recording** (JOIN **recording.artist** samuel_ramey))

what are the albums by Samuel Ramey?

(AND **music.album** (JOIN **album.artist** samuel_ramey))

Compositional Generalization

what are the albums by the artist who makes the recoding Holy Night?

(AND **music.album** (JOIN **album.artist**
(JOIN (R **recording.artist**) holy_night)))

Zero-Shot Generalization

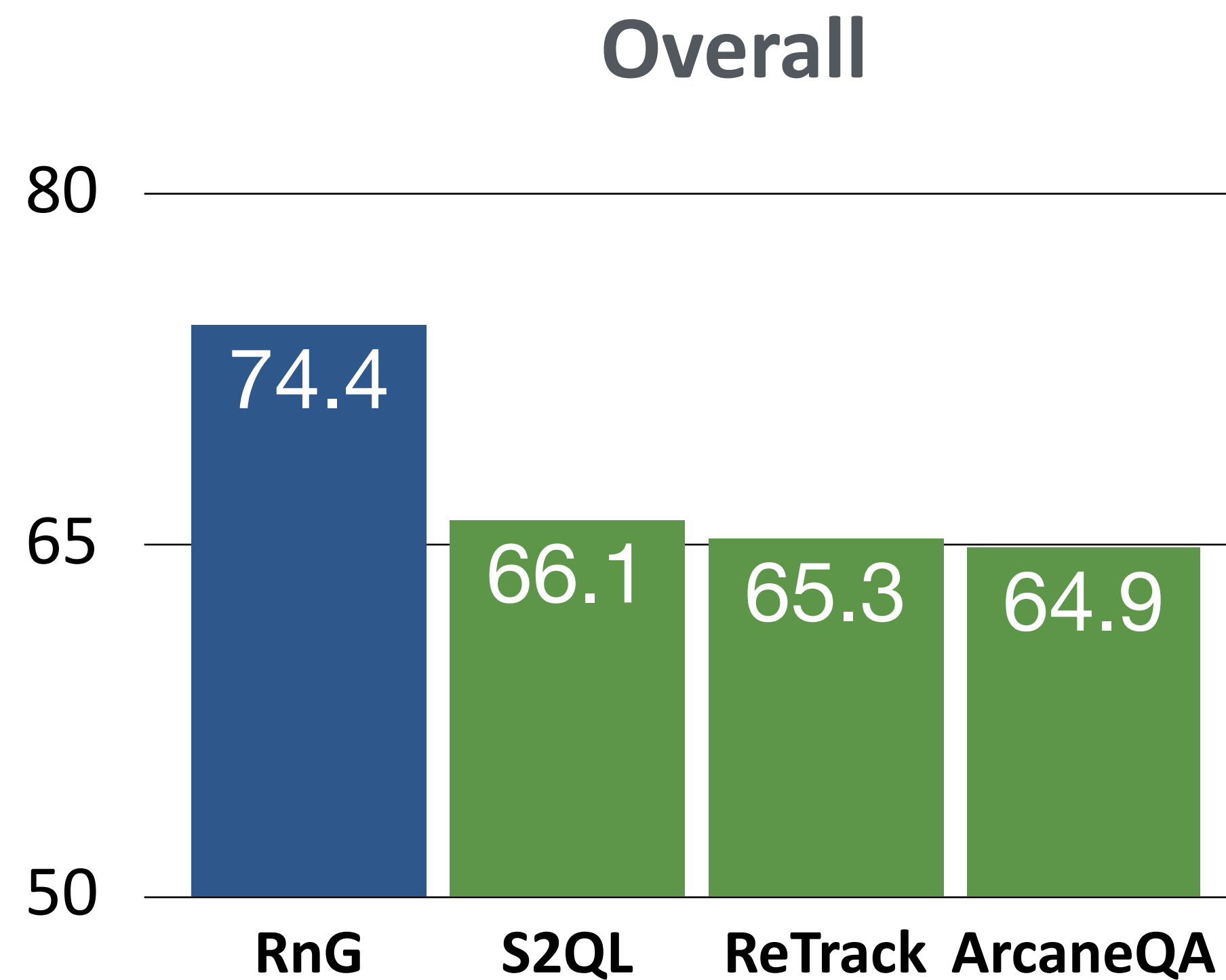
what songs for tv did Samuel Ramey write lyrics for?

(AND **tv.tv_song** (JOIN **composition.lyricist** samuel_ramey))



Results: GrailQA (Overall)

- ▶ RnG-KBQA achieves the **best overall performance**, surpassing other methods* by a large margin



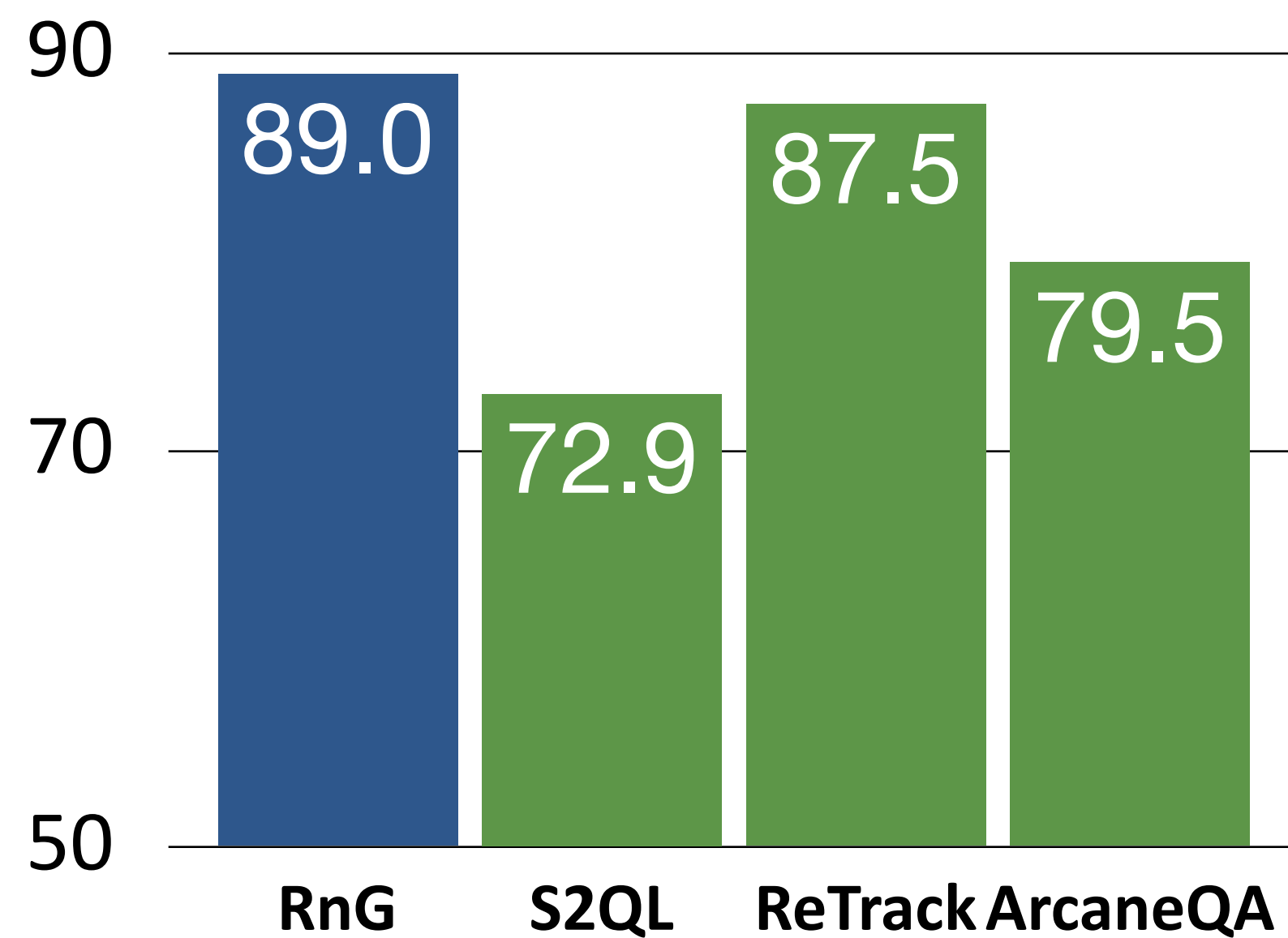
(* Results taken from the leaderboard as of 03/15/2022)



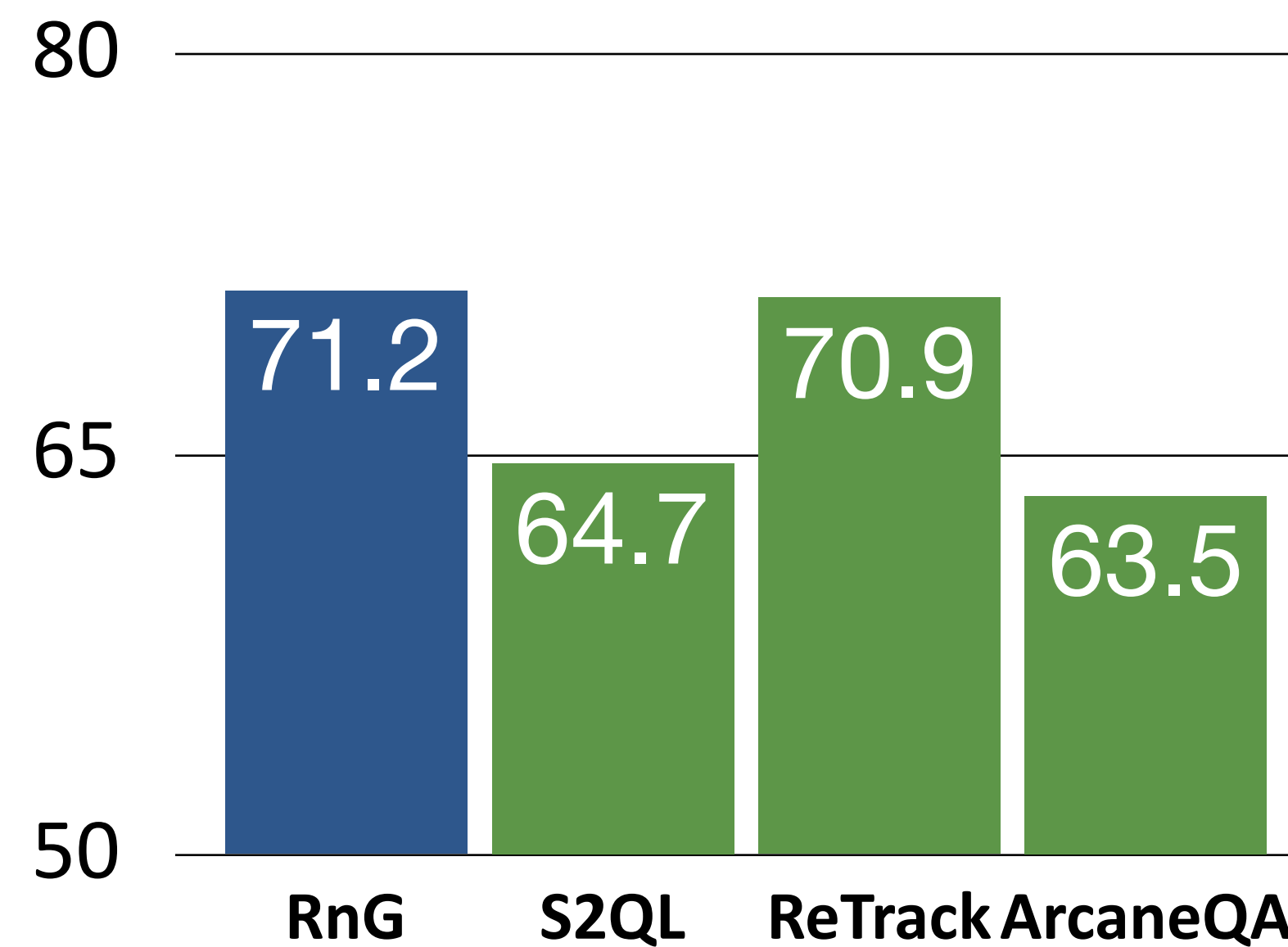
Results: GrailQA (Breakdown)

- ▶ RnG-KBQA is the best **across all generalization settings**, especially in zero-shot generalization

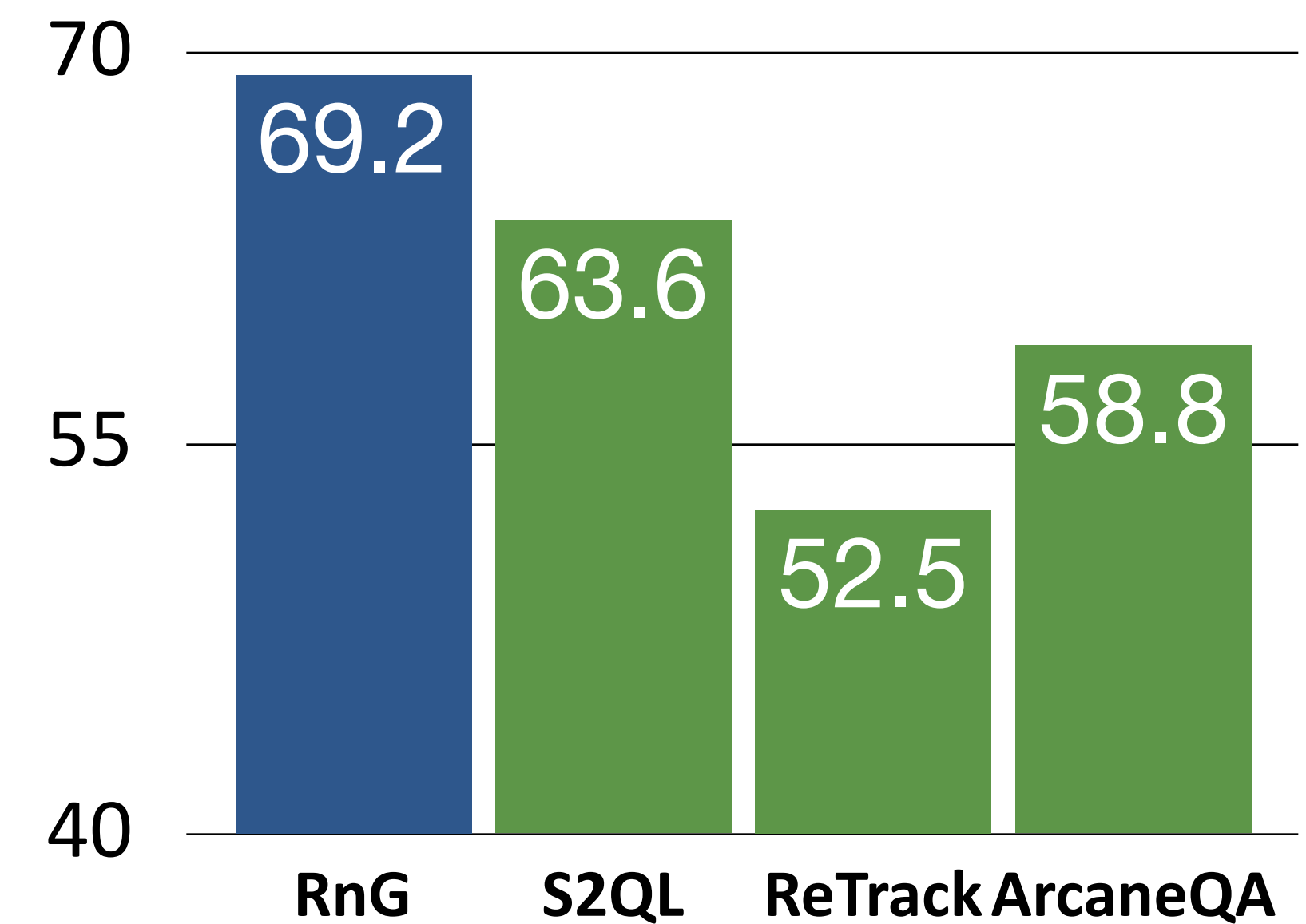
I.I.D.



Compositional

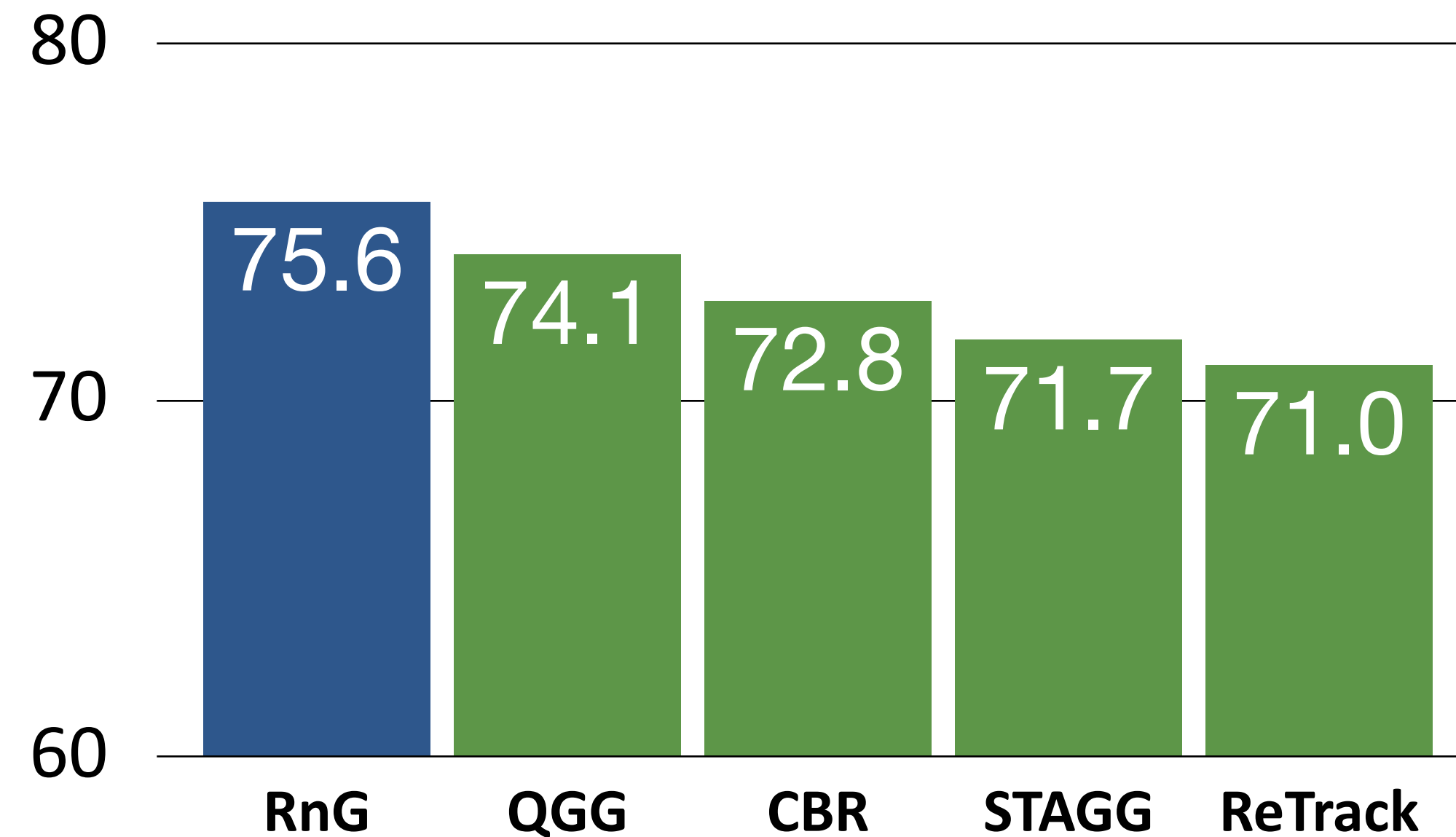


Zero-Shot



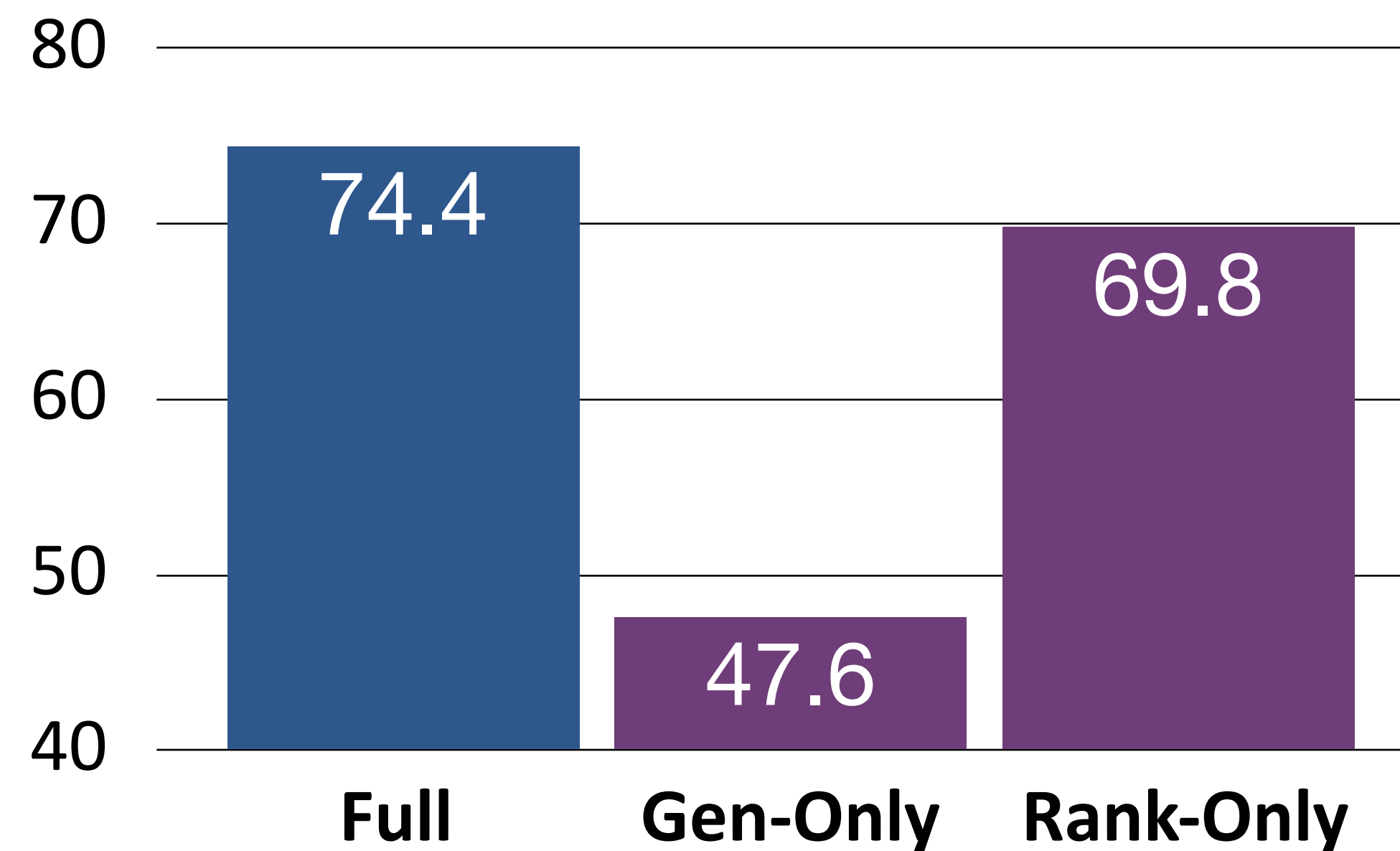
Results: WebQSP

- ▶ RnG-KBQA is still the best: our approach is effective across different datasets



Ablation Study

- ▶ **Full Model**
- ▶ **Generation-Only:** let the generation model conditioned on randomly sampled logical forms
- ▶ **Ranking-Only:** remove the generation model





Takeaways



Takeaways

- ▶ RnG-KBQA demonstrates strong performance across different generalization settings, and is especially strong in zero-shot generalization



Takeaways

- ▶ RnG-KBQA demonstrates strong performance across different generalization settings, and is especially strong in zero-shot generalization
- ▶ It is easier to assemble than to generate: we handle generalization problems by letting the generator be conditioned on useful components



Takeaways

- ▶ RnG-KBQA demonstrates strong performance across different generalization settings, and is especially strong in zero-shot generalization
- ▶ It is easier to assemble than to generate: we handle generalization problems by letting the generator be conditioned on useful components

Code Available at <https://github.com/salesforce/rng-kbqa>

