Learning to Rank Query Graphs for Complex Question Answering over Knowledge Graphs

G. Maheshwari, P. Trivedi, **D. Lukovnikov**, N. Chakraborty, A. Fischer, J. Lehmann

Outline

Introduction

Approach

Experiments

Future Work

QA over Knowledge Bases

Given:

a **natural language question**

a source **knowledge base**

Find the **answer** to the question using the given knowledge base.

Example: Simple Question

Name all Warner Brothers movies.

```
SELECT ?uri WHERE {
    ?uri dbp:studio dbr:Warner_Bros .
}
```

Complex

How many Marvel movies was Robert Downey Jr. casted in?

```
SELECT COUNT(?uri) WHERE {
    ?uri dbp:studio dbr:Marvel_Studios.
    ?uri dbp:starring dbr:Robert_Downey_Jr
}
```

Question Answering is Hard!

Name all the movies in which Robert Downey Jr acted?
Which movies have RDJ?
Flicks where I can see Robert DJ?
Find me all the films casting Robert Downey Jr ?
List all the movies starring Robert Downey Junior?
RDJ has acted in which movies?

```
SELECT ?uri WHERE {
    ?uri dbp:starring dbr:Robert_Downey_Jr .
}
```

* rdf:type constraint omitted for brevity's sake

QA as NL to SPARQL

QA over KG can be solved by converting **natural language question to SPARQL queries**.

~Translation problem

General approach

We follow the general approach which

- Creates a set of formal query candidates
- Learn to rank them with respect to Question

Approach

Overview

```
Give me a list of everything where Robert Downey Jr Acted?
```

Overview

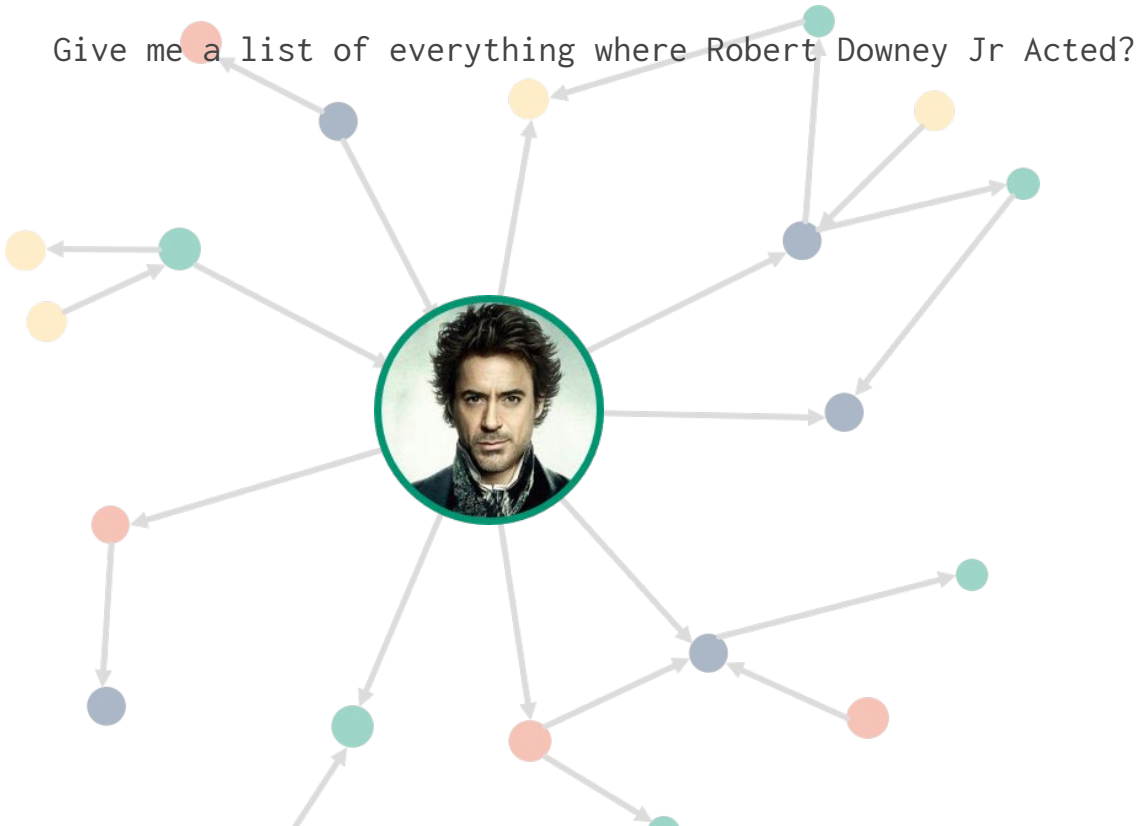
Given: Question, Topic Entity.

`Give me a list of everything where` **`Robert Downey Jr`** `Acted?`

Overview

Given: Question, Topic Entity.

Collect 2-hop subgraph around it.



Overview

Given: Question, Topic Entity.

Collect 2-hop subgraph around it.

Generate core-chain candidates

```
Give me a list of everything where Robert Downey Jr Acted?
```

```
dbr:Robert_Downey_Jr + dbp:birthplace
dbr:Robert_Downey_Jr + dbp:parent
dbr:Robert_Downey_Jr + dbp:spouse - dbp:foundedBy
dbr:Robert_Downey_Jr - dbp:starring
dbr:Robert_Downey_Jr - dbp:starring + dbp:director
...
```

Overview

Given: Question, Topic Entity.

Collect 2-hop subgraph around it.

Generate core-chain candidates

Rank Candidates based on similarity with questions

```
Give me a list of everything where Robert Downey Jr Acted?

0.10 dbr:Robert_Downey_Jr + dbp:birthplace
0.23 dbr:Robert_Downey_Jr + dbp:parent
0.04 dbr:Robert_Downey_Jr + dbp:spouse - dbp:foundedBy
0.73 dbr:Robert_Downey_Jr - dbp:starring
0.41 dbr:Robert_Downey_Jr - dbp:starring + dbp:director
     ...
```

How to represent SPARQL

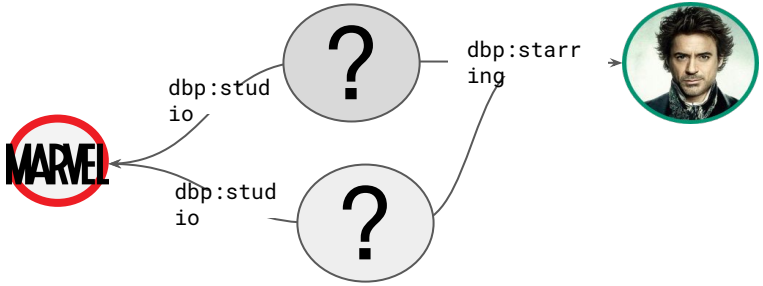
```
SELECT ?uri WHERE {
    ?uri dbp:studio dbr:Marvel_Studios.
    ?uri dbp:starring dbr:Robert_Downey_Jr
}
```

How to represent SPARQL

```
?uri dbp:studio dbr:Marvel_Studios.
?uri dbp:starring dbr:Robert_Downey_Jr
```

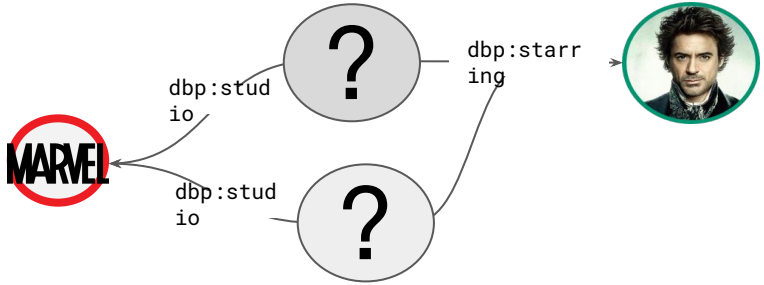
How to represent SPARQL

```
?uri dbp:studio dbr:Marvel_Studios.
?uri dbp:starring dbr:Robert_Downey_Jr
```



Core chain

```
?uri dbp:studio dbr:Marvel_Studios.
?uri dbp:starring dbr:Robert_Downey_Jr
```



dbr:Marvel_Studios - dbp:studio + dbp:starring dbr:Robert_Downey_Jr

Core chain

dbp:Marvel_Studios - dbr:studio + dbo:starring dbr:Robert_Downey_Jr

Core chain- dropping entities

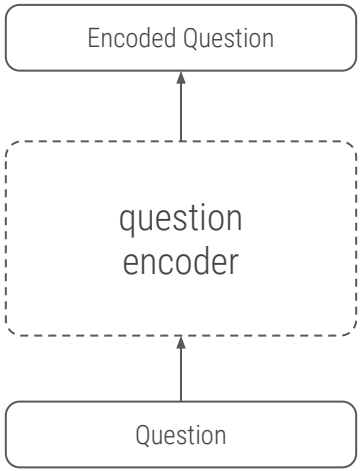
- `dbr:studio` + `dbo:starring`

Approach

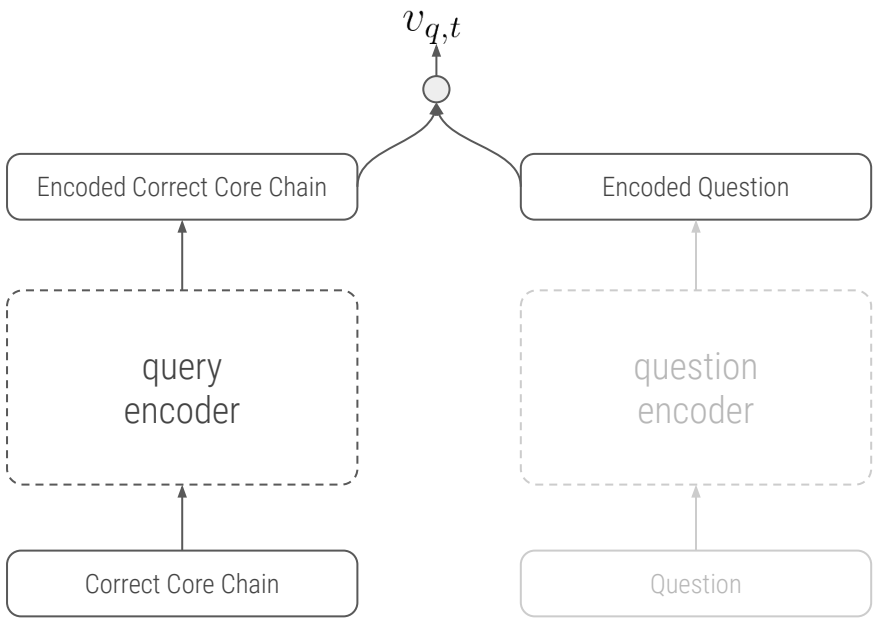
Ranking Framework

Encode core chain and question to a **vector space** such that **distance** between the correct core chain and the question is **smaller**.

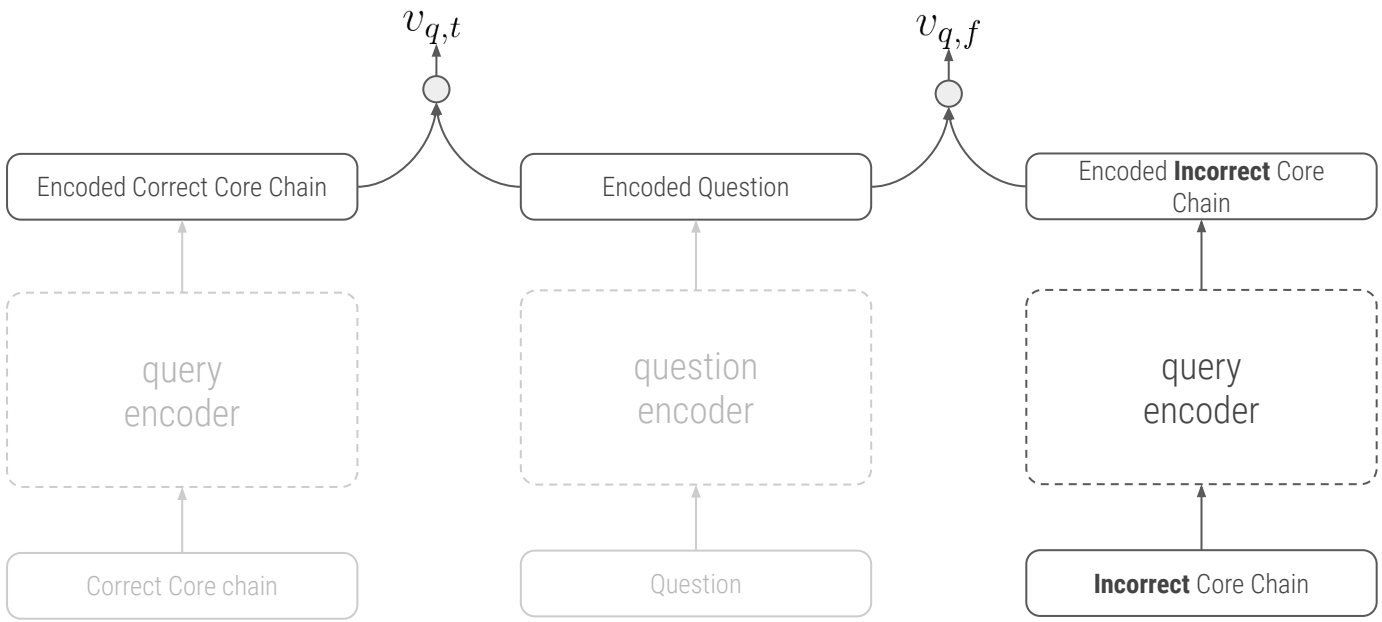
Ranking Framework



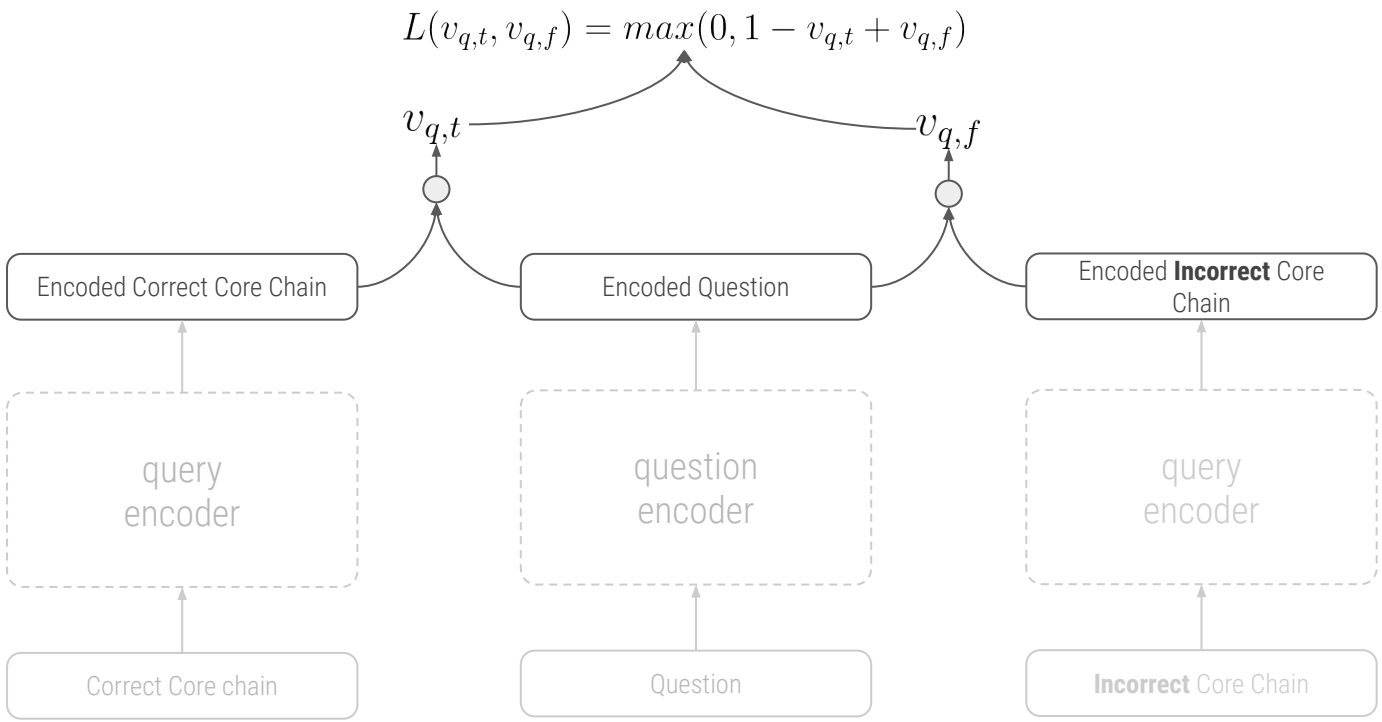
Ranking Framework



Ranking Framework - Training

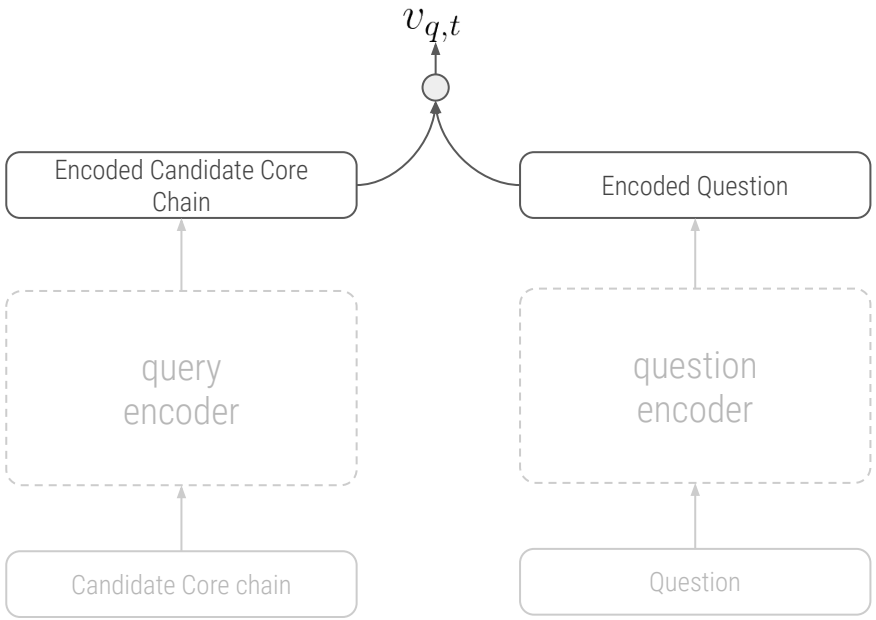


Ranking Framework - Training



Ranking Framework - Prediction

1. Generate possible candidates given the entities
2. Rank them using model
3. Take highest scoring one



Additional Constraints

Information needed to reconstruct SPARQL:

1. Triple BGP
    a. Via core chain
2. Fact retrieval (SELECT) /Fact Validation (ASK)/Fact Counting (SELECT COUNT)
    a. Via Classifier
3. Additional Type Constraints
    a. Ranking Model (BiLSTM)

Approach

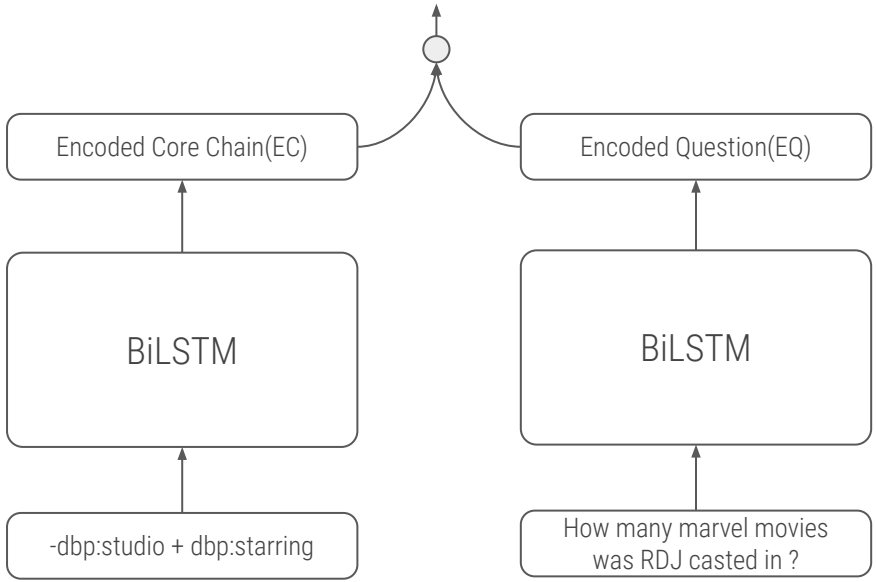
Ranking Model

We use **neural networks** to encode question as well as candidates.

Several encoders

- BiLSTM [4]
- Decomposable Attention [5]
- CNN [6]
- Transformers [7]

BiLSTM Encoder



BiLSTM Encoder Problems

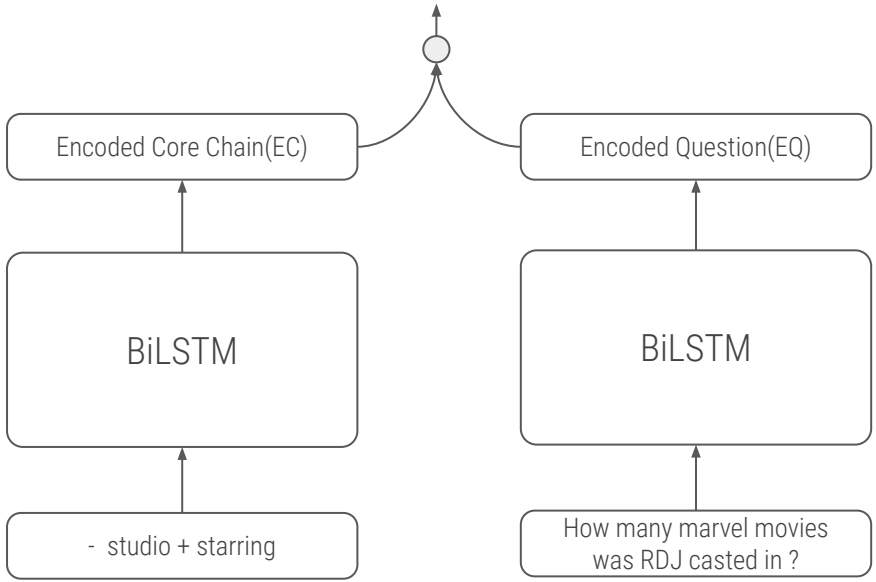
Forcing the model

- two predicates in the corechain in same vector.
- One question representation to include information about both the predicates in the question.

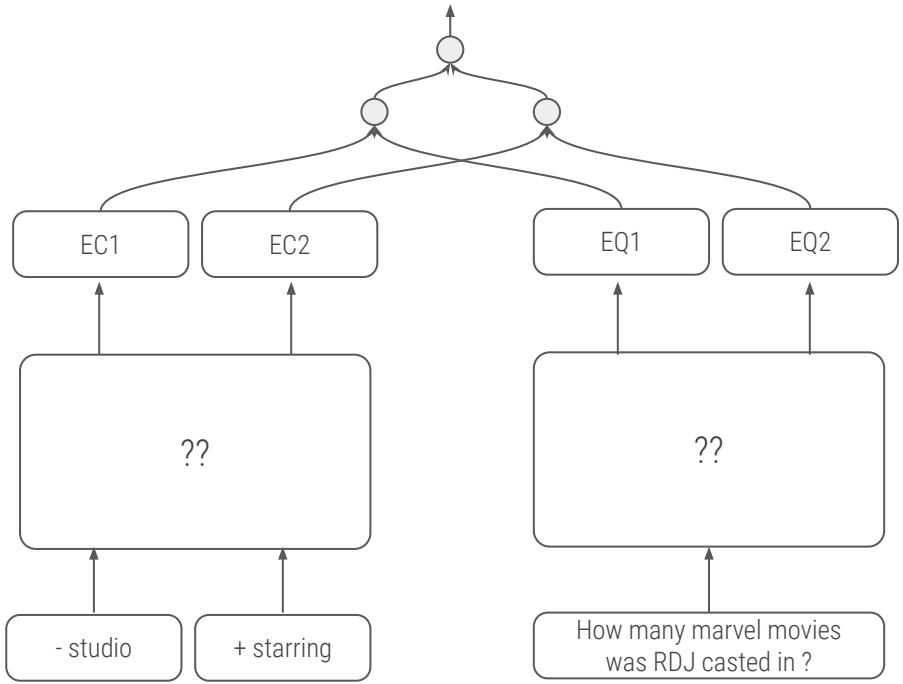
Solutions

- Encode the predicates in the core chain individually
- Creating two encodings of questions, for each relation "slot"
- Compare the two

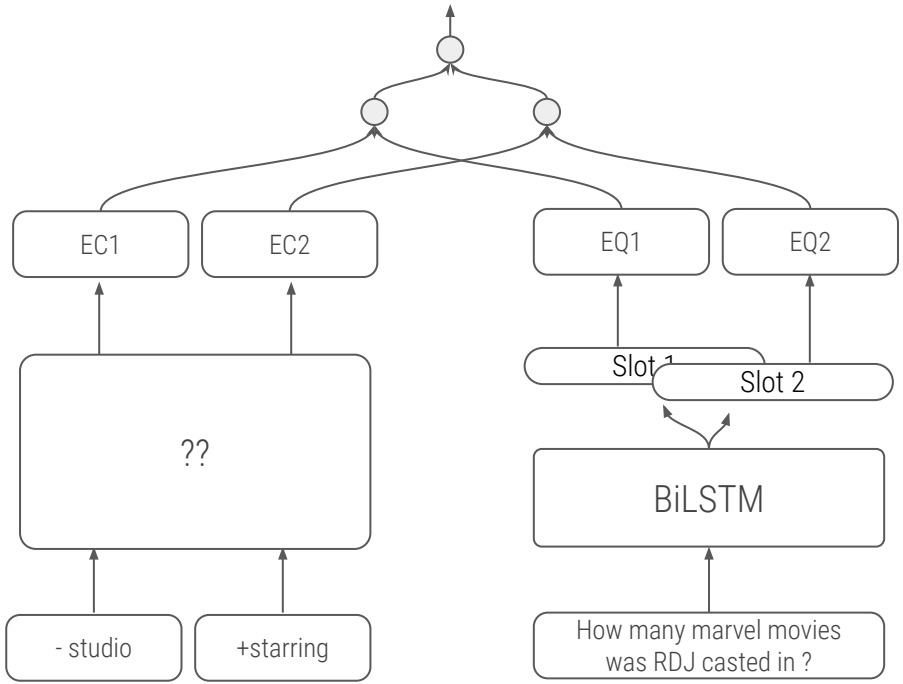
BiLSTM Encoder



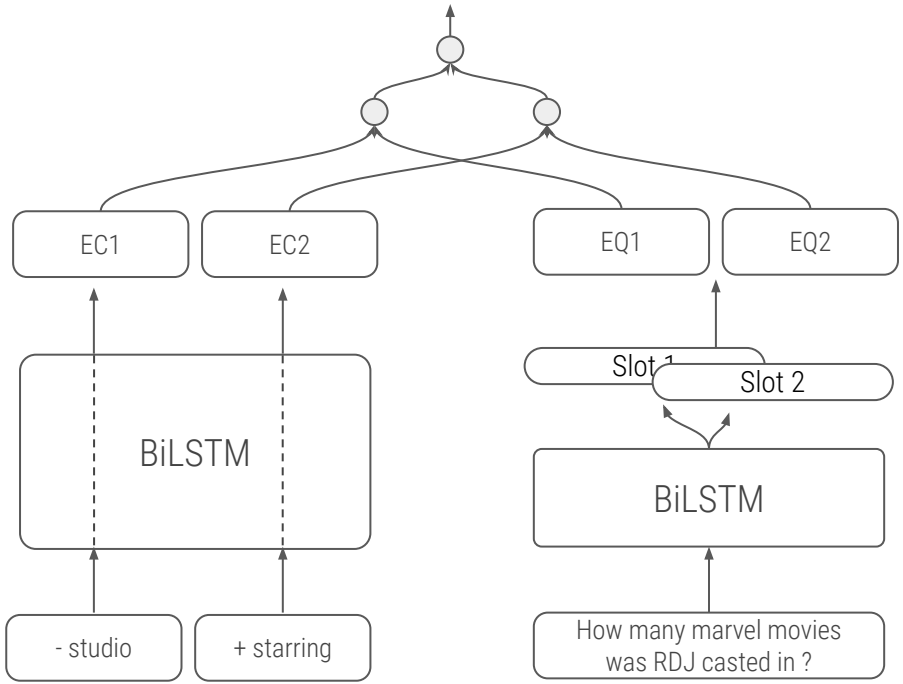
Updated BiLSTM Encoder



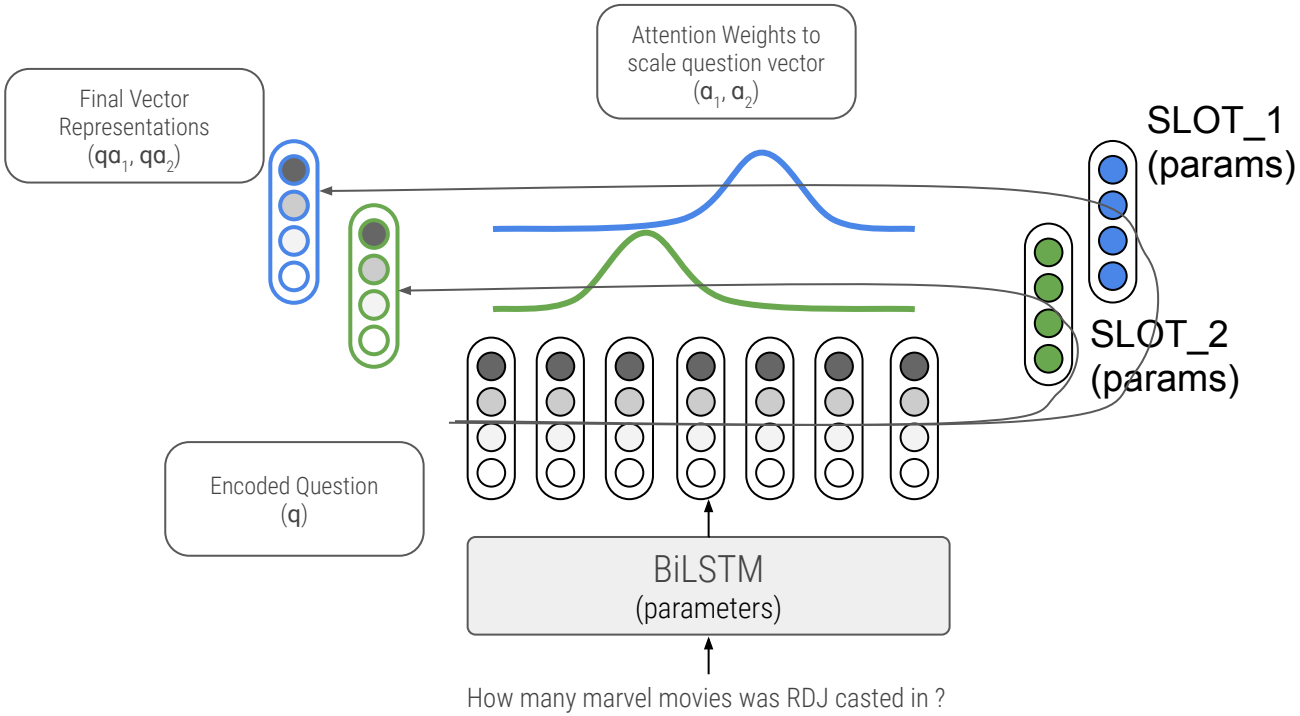
Slot - Matching Encoder



Slot - Matching Encoder



Slot Attention



Experiments

Datasets

LC-QuAD [8] - 5000 question-sparql pairs.

QALD-7 [9] - 263 question-sparql pairs.

Both are based on DBpedia 2016-04 release.

Experiments

Baselines

We experiment with different encoder architectures

- BiLSTM [4]
- Decomposable Attention (DAM) [5]
- Multi-channel Convolutional Neural Network (CNN) [6]
- Hierarchical Residual Model (HRM) [10]

Results

Results of different models on LC-QuaD and QALD

Reporting **core chain accuracy** (CCA)

(see paper for F1, Precision, Recall of execution results)

Models	LC-QuAD CCA	QALD CCA
BiLSTM	0.61	0.28
CNN	0.44	**0.31**
HRM	0.62	0.28
DAM	0.57	0.28
Slot Matching (BiLSTM)	**0.63**	**0.31**

Slot matching analysis

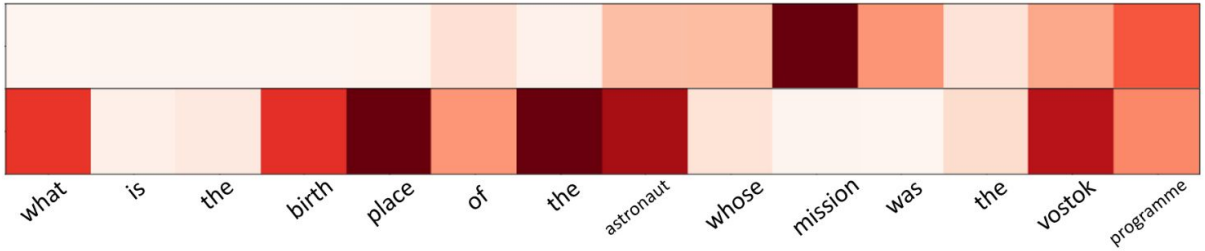
Attention visualization of Slots.

The two rows represents different slot attention scores

dbr:vostok_programmee

dbo:mission

dbo:birthPlace



Experiments

Transfer Learning

1. Pretrain on LC-QuAD and fine tune on QALD.

2. From pretrained language model (BERT [11]).

Transfer between datasets

Train on LC-QuAD data and then finetune on QALD

Different learning rate schedules while fine-tuning

Constant - LR remains same throughout the training.

Slanted Triangular - Linearly increase until one point, and then linearly decrease.

Cosine with restarts - LR follows a cosine curve throughout the training process.

Transfer between datasets

Learning Rate	BiLSTM	Slot Matching
constant	0.37	0.37
Slanted triangular	**0.39**	**0.42**
Cosine with warm restarts	0.25	0.28
Without pretraining	0.28	0.31

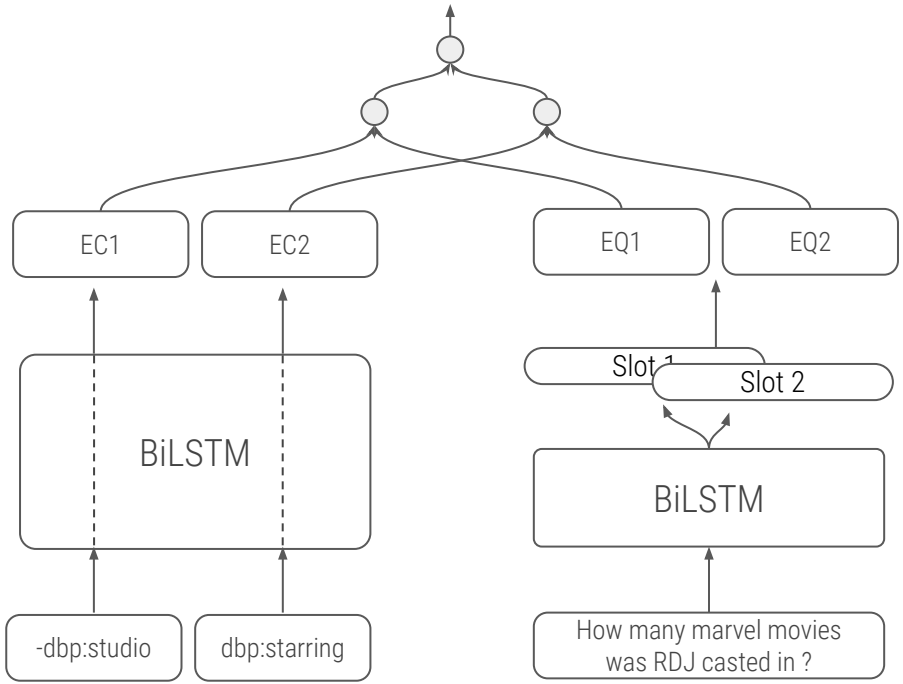
Pre-trained on LC-QuaD, fine-tuned on QALD

Numbers tested for QALD.

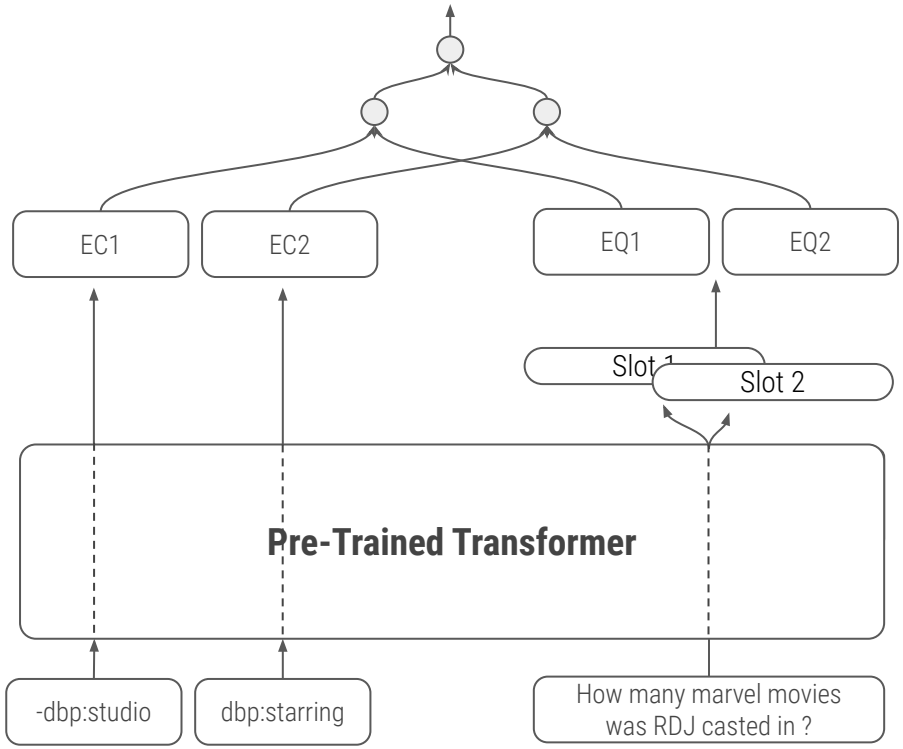
Pre Training on Language Model

Replace the Bi-LSTM part of the architecture with pre-trained Transformer (BERT)

BiLSTM (Slot Matching)



Pre Training (Slot Matching)



Pre Trained Language Model

	QALD	LC-QuAD
Bert	0.23	0.67
Bert Slot matching	0.18	**0.68**
BiLSTM	0.28	0.61
Slot Matching	**0.31**	0.63

Conclusion

Conclusion

Slot pointer seems to help.

Transfer Learning helps.

Thank You

References

[1] Lehmann, Jens, et al. "DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia." Semantic Web 6.2 (2015): 167-195.

[2] Bollacker, Kurt, et al. "Freebase: a collaboratively created graph database for structuring human knowledge." Proceedings of the 2008 ACM SIGMOD international conference on Management of data. AcM, 2008.

[3] Vrandečić, Denny, and Markus Krötzsch. "Wikidata: a free collaborative knowledge base." (2014).

[4] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[5] Parikh, Ankur P., et al. "A decomposable attention model for natural language inference." arXiv preprint arXiv:1606.01933 (2016).

[6] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[7] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

[8] Trivedi, Priyansh, et al. "Lc-quad: A corpus for complex question answering over knowledge graphs." International Semantic Web Conference. Springer, Cham, 2017.

[9] Usbeck, Ricardo, et al. "7th open challenge on question answering over linked data (QALD-7)." Semantic Web Evaluation Challenge. Springer, Cham, 2017.

[10] Yu, Mo, et al. "Improved neural relation detection for knowledge base question answering." arXiv preprint arXiv:1704.06194 (2017).

[11] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

References (Icons)

Sherlock holmes by Matthew Davis from the Noun Project

Empire State Building by Jake Dunham from the Noun Project

Golden gate bridge by icon 54 from the Noun Project

Statue of Liberty by Berkah Icon from the Noun Project

MARVEl is from flaticon

Sherlock Holmes is from FlatIcon

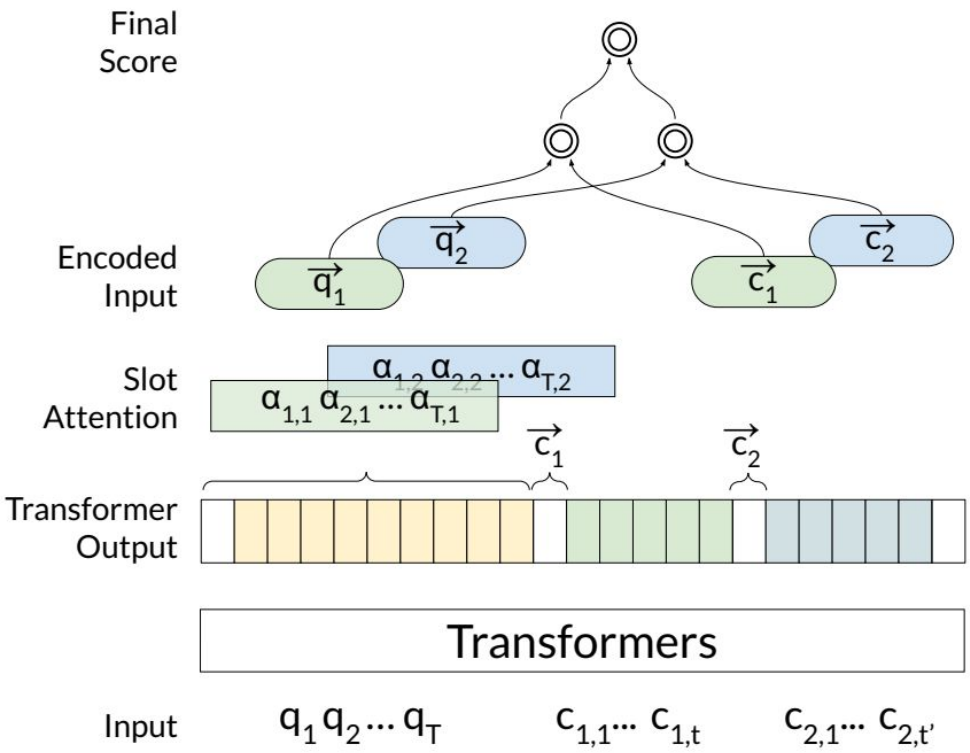
Iron Man by Tatyana Kyul from the Noun Project

Ferguson by priyanka from the Noun Project

Appendix

Core Chain to SPARQL

Template Matching

+ relation$_1$ | entity$_1$	{ ?uri relation$_1$ entity$_1$ }
- relation$_1$ + relation$_2$ | entity$_1$	{?x relation$_1$?uri ?x relation$_2$ entity$_1$ }
entity$_1$ | relation$_2$ - relation$_1$ | entity$_2$	{ ?uri relation$_2$ entity$_1$ entity$_2$ relation$_1$?uri}

Pre Training (Slot Matching Pre-Trained)