



Build a **Question Answering** system overnight

Jens Lehmann, Gaurav Maheshwari, Priyansh Trivedi, Mohnish Dubey, Denis Lukovnikov,

ESWC 2019, Portoroz

Framework Recap

Overview

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

Overview

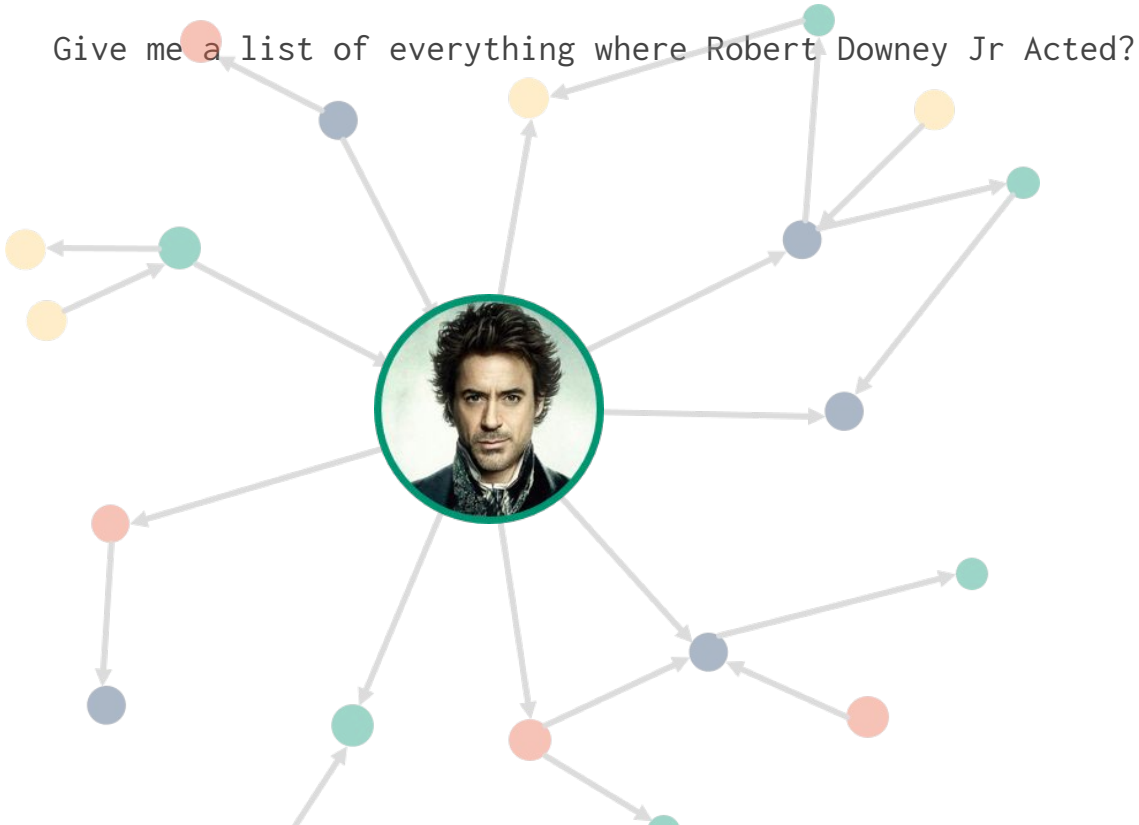
Given: Question, Topic Entity.

`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?
```

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

Ranking Framework

Encode core chain and question to a **vector space** such that the correct core chain and the question are **aligned** with one another.

Ranking Framework

Encode core chain and question to a **vector space** such that the correct core chain and the question are **aligned** with one another.

$$\text{ans} = \underset{c}{\text{argmax}}\Big(\text{compare}\big(\text{enc}_q(q), \text{enc}_c(c)\big)\Big)$$

Question Answering Framework

Ranking Framework

1. **Encode**
2. **Compare**
3. **Select**

$$\text{ans} = \underset{c}{\text{argmax}} \Big(\text{compare}(\text{enc}_q(q), \text{enc}_c(c)) \Big)$$

1. Encode

Encoding is a function mapping inputs from one space to another.

$$\text{enc} : \mathcal{X}_a \mapsto \mathcal{X}_b$$

1. Encode $\text{enc} : \mathcal{X}_a \mapsto \mathcal{X}_b$

$\mathcal{X}_a$ is a space of text.

{‘who is’ , ‘president’, ‘river rhine’, ‘potato’ …. } $\subseteq \mathcal{X}_a$

$\mathcal{X}_b$ should be a space where the correct core chain corresponding to a question should be closest to it

1. Encode $\mathrm{enc} : \mathcal{X}_a \mapsto \mathcal{X}_b$

We'll come back to this, later.

For now,

$$\vec{q} = \mathrm{enc}(q)$$
$$\vec{c} = \mathrm{enc}(c)$$

2. Compare

We need a mechanism to *judge the closeness* between $\vec{q}$, $\vec{c}$

2. Compare

We need a mechanism to *judge the closeness* between $\vec{q}$, $\vec{c}$

$$score = \vec{q}.\vec{c}$$

3. Select

Given multiple **candidates** for the correct core chain, we use the $score$ to select the correct one.

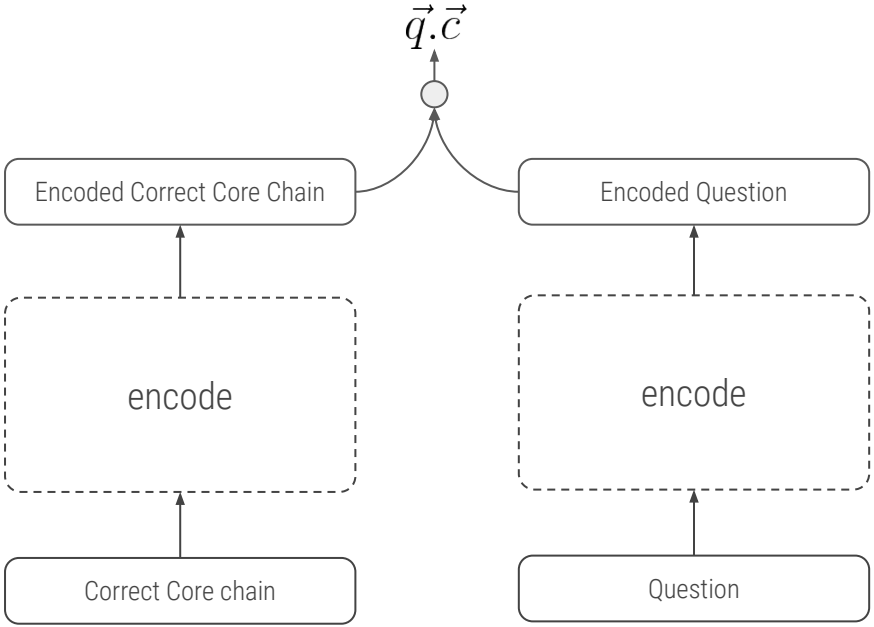
$$\text{answer} = \underset{c}{\text{argmax}}\ \vec{q}.\vec{c}$$

Ranking Framework

1. **Encode**
2. **Compare**
3. **Select**

$$\text{ans} = \underset{c}{\text{argmax}} \Big(\text{compare}(\text{enc}_q(q), \text{enc}_c(c)) \Big)$$

Ranking Framework



How do we accomplish this?

Encoding to space where the correct core chain corresponding to a question should be closest to it

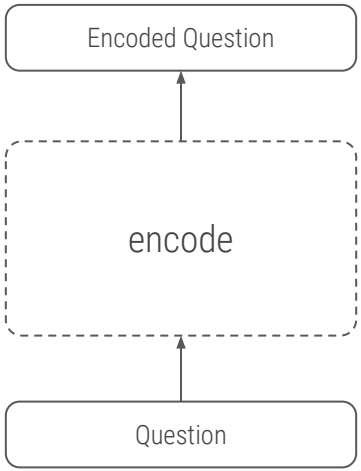
Training
Neural Ranking Models

Recall

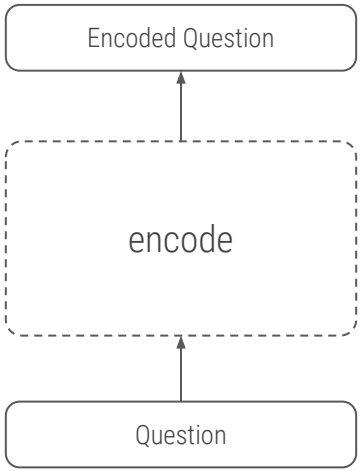
q Give me a list of everything where Robert Downey Jr Acted?

c + dbp:birthplace
+ dbp:parent
+ dbp:spouse - dbp:foundedBy
- dbp:starring
- dbp:starring + dbp:director
...

Ranking Framework

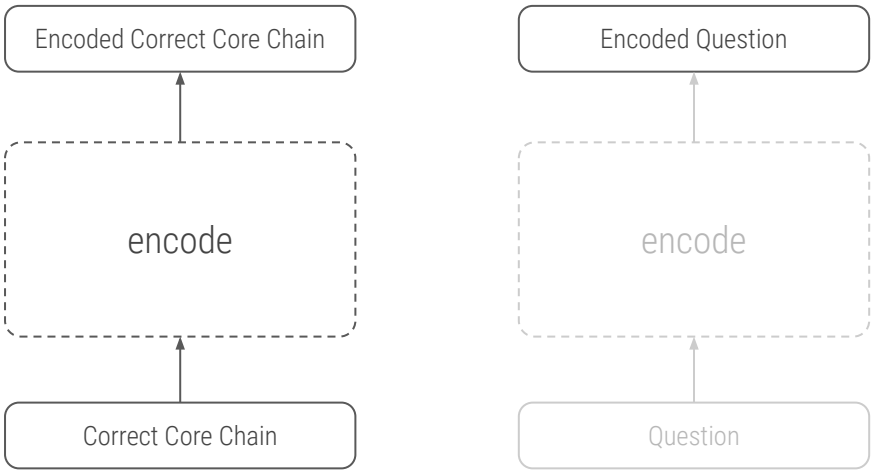


Ranking Framework

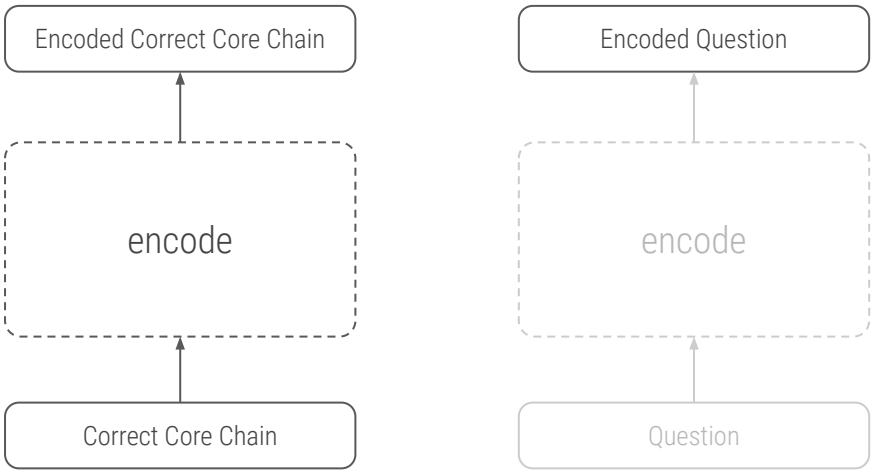


Give me a list of everything where Robert Downey Jr Acted?

Ranking Framework

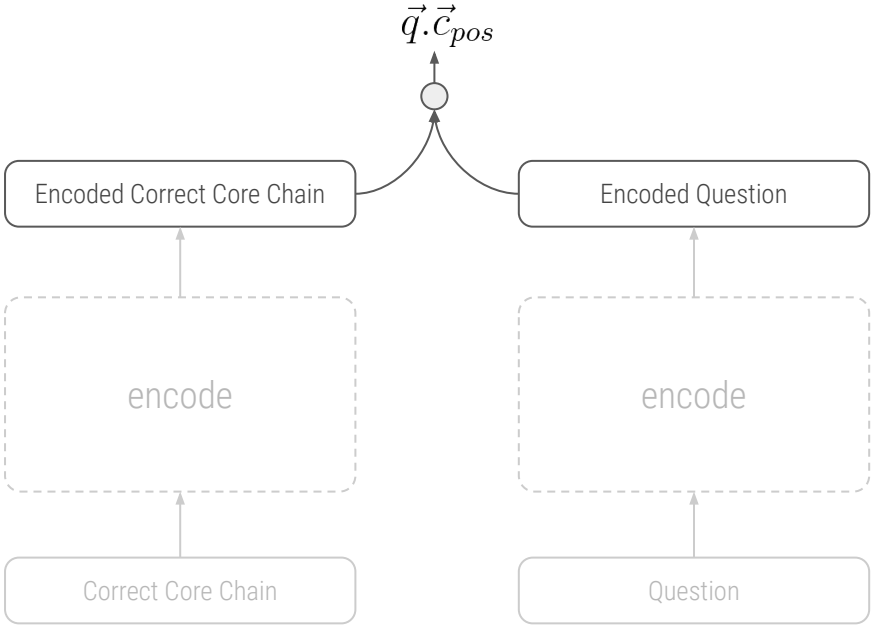


Ranking Framework



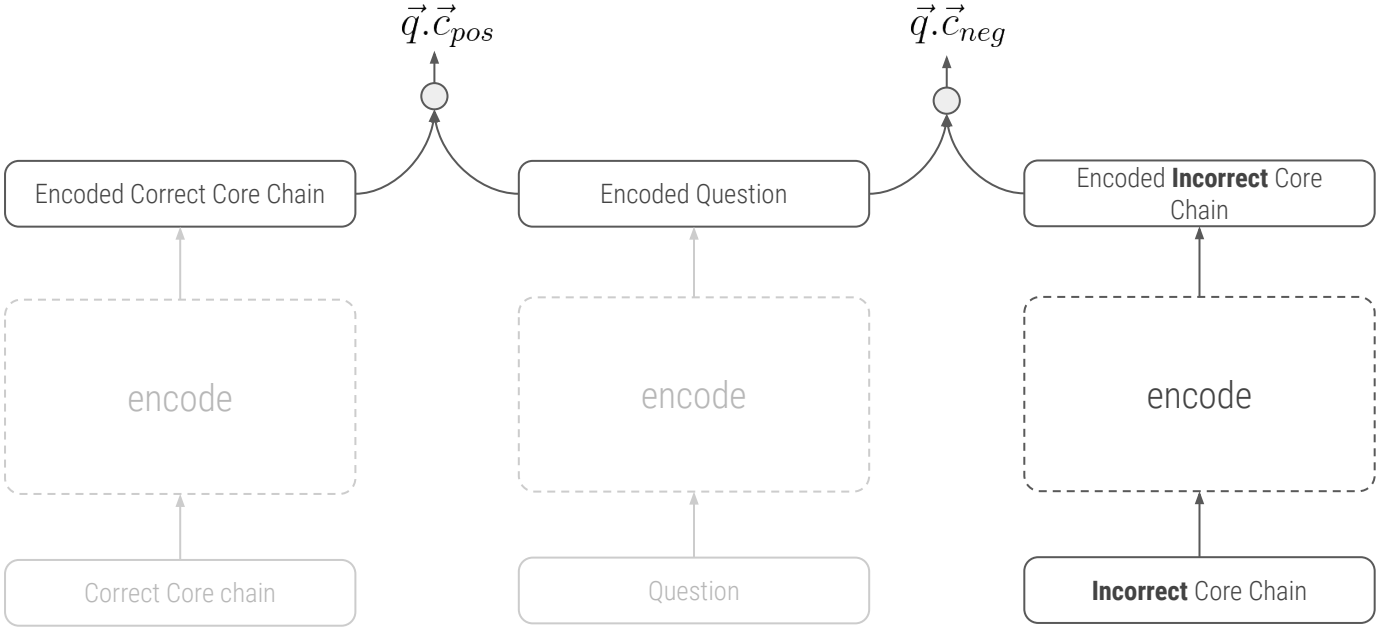
- dbp:starring

Ranking Framework

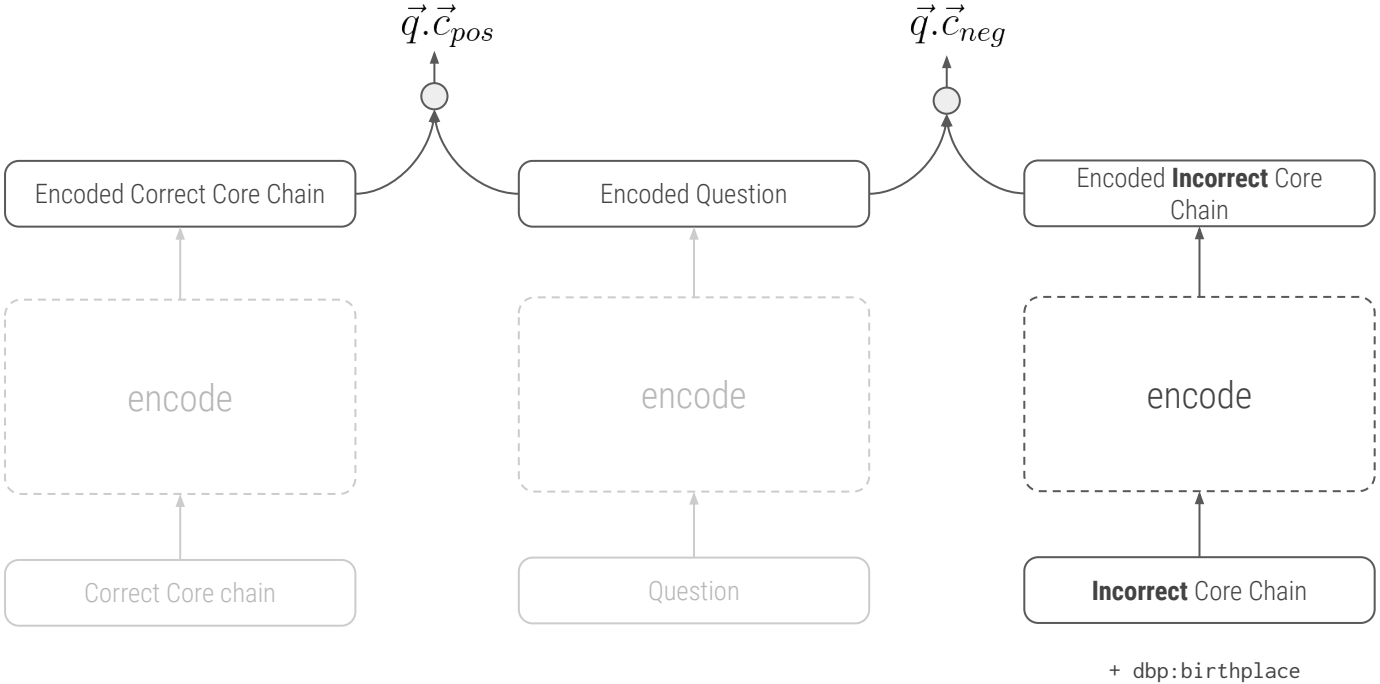


- dbp:starring

Ranking Framework

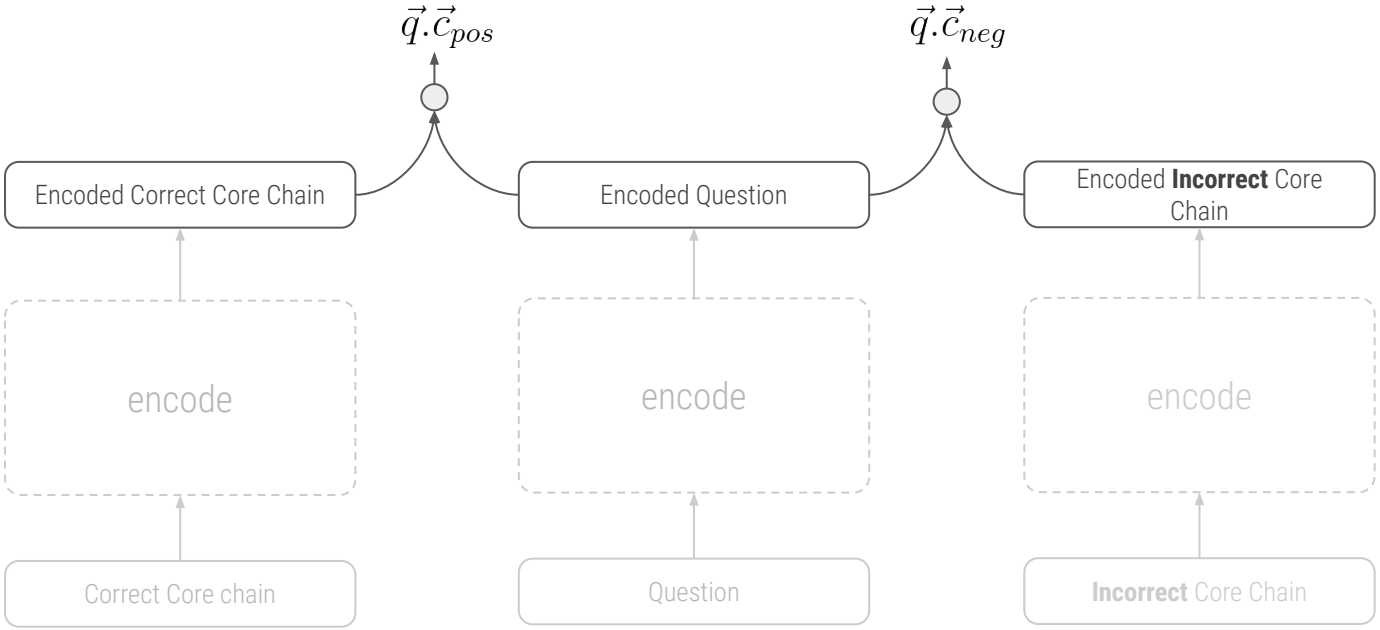


Ranking Framework



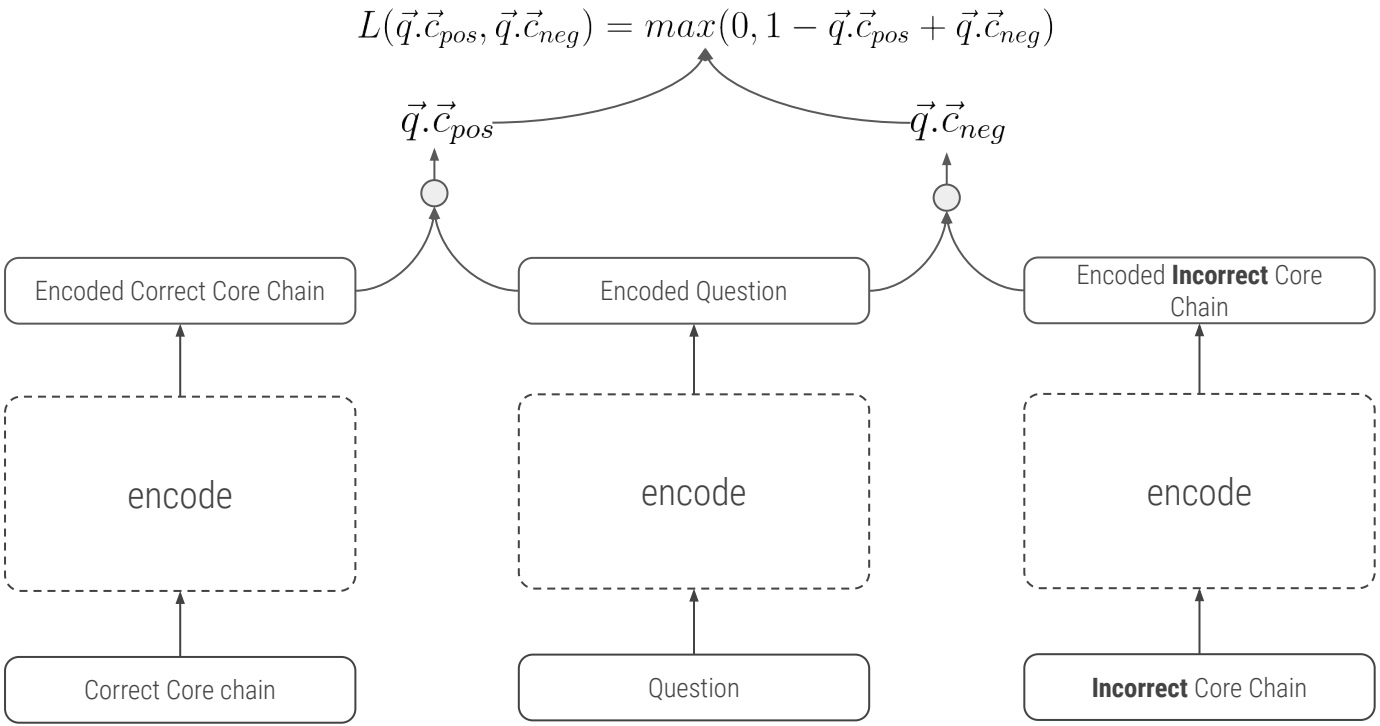
+ dbp:birthplace

Ranking Framework



- dbp:starring Give me a list of everything where Robert Downey Jr Acted? + dbp:birthplace

Ranking Framework



Training Steps

$$L(\vec{q}.\vec{c}_{pos}, \vec{q}.\vec{c}_{neg}) = max(0, 1 - \vec{q}.\vec{c}_{pos} + \vec{q}.\vec{c}_{neg})$$

Given a set of questions, and SPARQL queries, for every (question, query) pair:

1. Find the entity
2. Collect core chain candidates
3. Parse the query to get the correct core chain candidates
4. Sample k different incorrect core chains, and pass it to the loss function above to train.

Programming