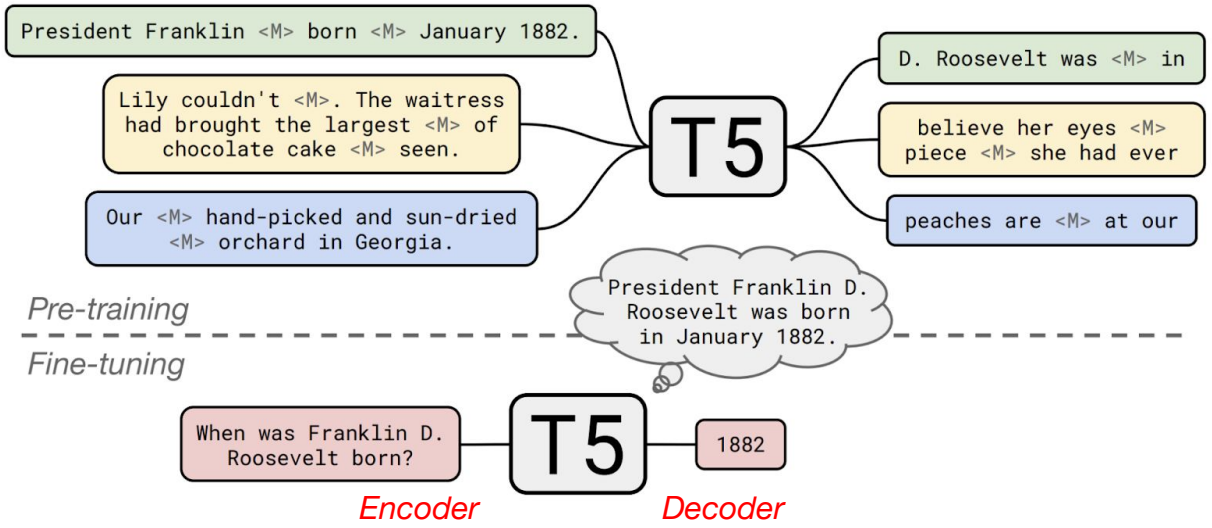
Language Models as Structured KBs

Bhuwan Dhingra

Duke
NLP

(Masked) Language Modeling



How much knowledge can you pack into the parameters of a language model? *Roberts et al, 2020*

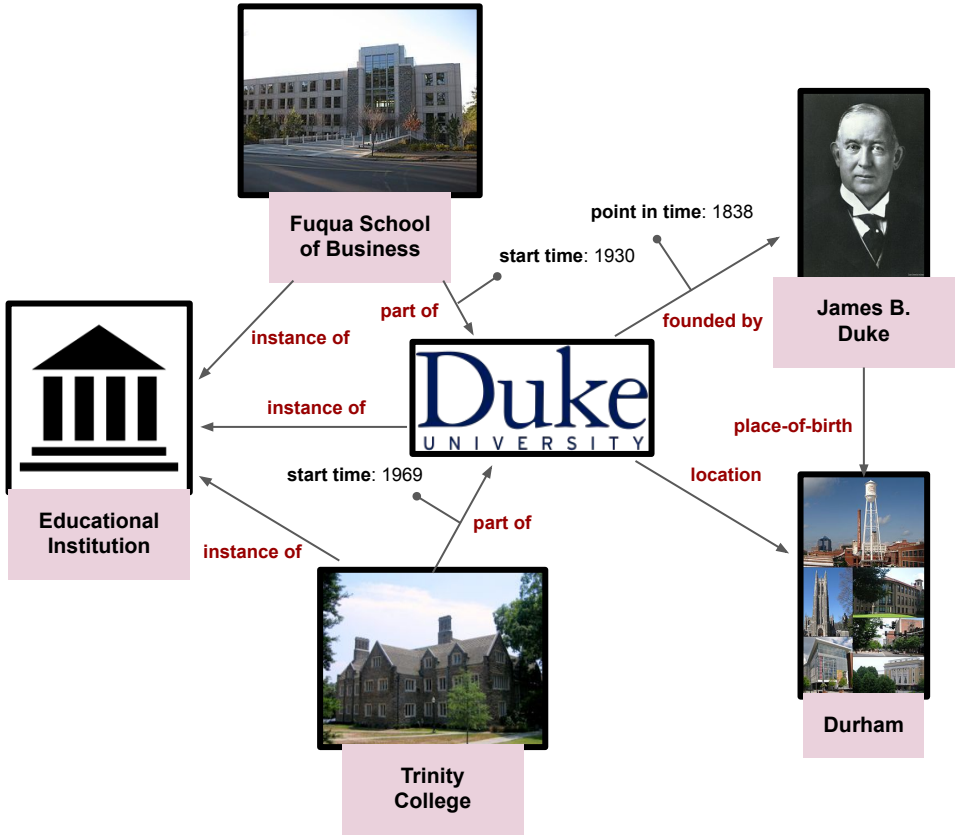
Knowledge Graphs

```
SELECT ?x
WHERE {
      "Duke University" founded-by ?y
      ?y place-of-birth ?x
}
```

"Place of birth of the founder of Duke University"

```
SELECT ?x
WHERE {
      ?x instance-of "Education Inst"
      ?x part-of ?property
      ?property object "Duke University"
      ?property start-time ?start
      FILTER { ?start < 1945 }
}
```

"Duke colleges as of 1945"



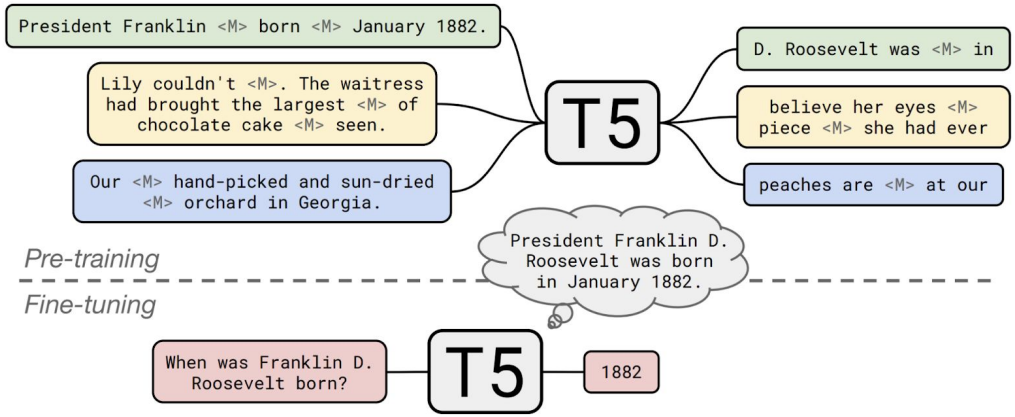
Can we do this with language models?

```
SELECT ?x
WHERE {
      "Duke University" founded-by ?y
      ?y place-of-birth ?x
}
```

"Place of birth of the founder of Duke University"

```
SELECT ?x
WHERE {
      ?x instance-of "Education Inst"
      ?x part-of ?property
      ?property object "Duke University"
      ?property start-time ?start
      FILTER { ?start < 1945 }
}
```

"Duke colleges as of 1945"



Overview

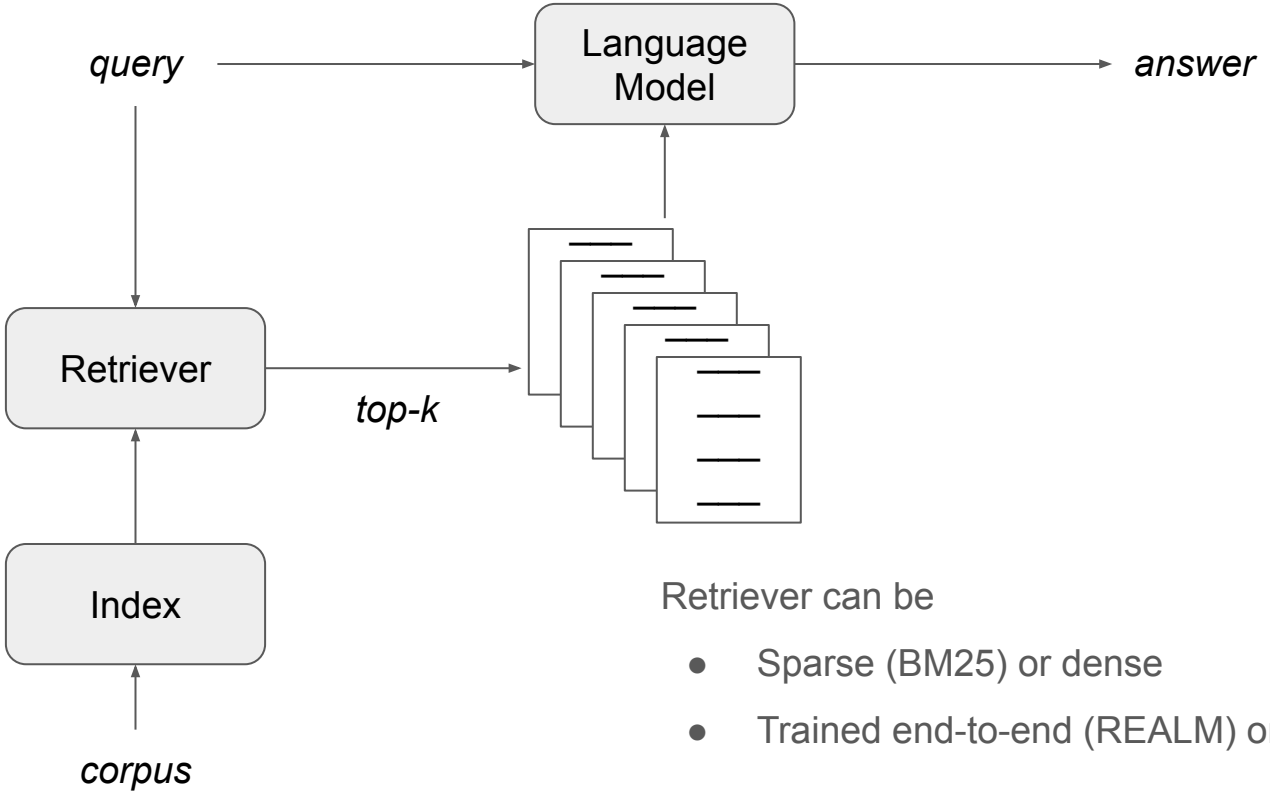
1. Differentiable query language over text
   - ➢ Differentiable Reasoning over a Virtual KB *Dhingra et al, ICLR 2020*
   - ➢ Reasoning Over Virtual KBs With Open Predicate Relations *Sun et al, ICML 2021*

2. Adding temporal scopes to pretrained knowledge inside LMs
   - ➢ Time-Aware LMs as Temporal KBs *Dhingra et al, 2021 (Under Review)*

Setup: Factoid Question Answering

- Given a query *q* and a corpus *C* we need to find an entity answer *a*
- Assume access to an entity linking system over entities *E*

Retrieval augmented models

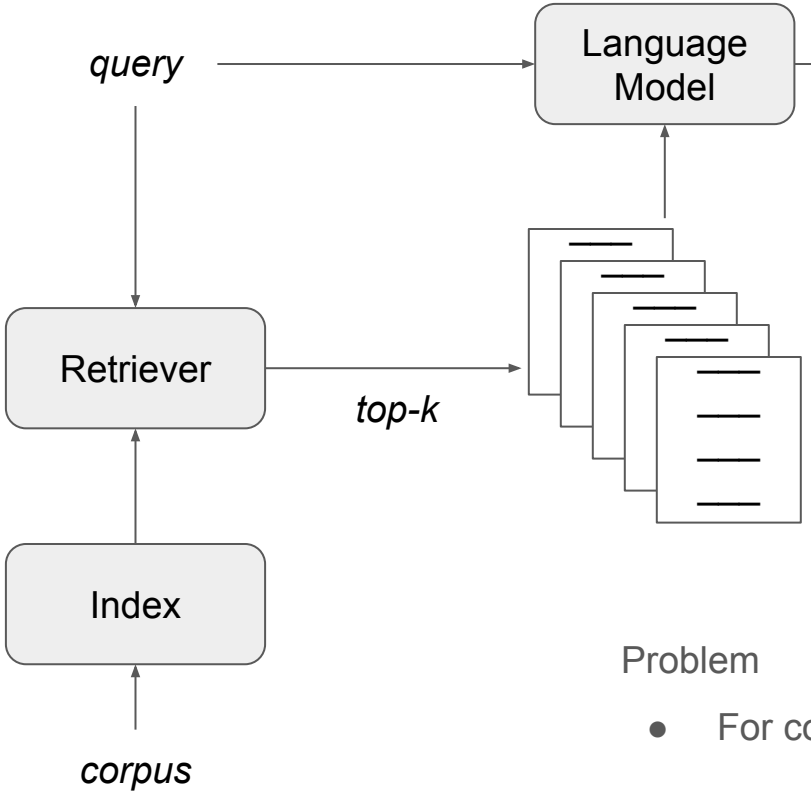


Retriever can be

- Sparse (BM25) or dense
- Trained end-to-end (REALM) or separately (DPR)

REALM: Retrieval-Augmented Language Model Pre-Training *Guu et al, 2020*
Dense Passage Retrieval for Open-Domain Question Answering *Karpukhin et al, 2020*

Retrieval augmented models




```
SELECT ?x
WHERE {
       "Duke University" founded-by ?y
       ?y place-of-birth ?x
}
```

"Place of birth of the founder of Duke University"

Problem

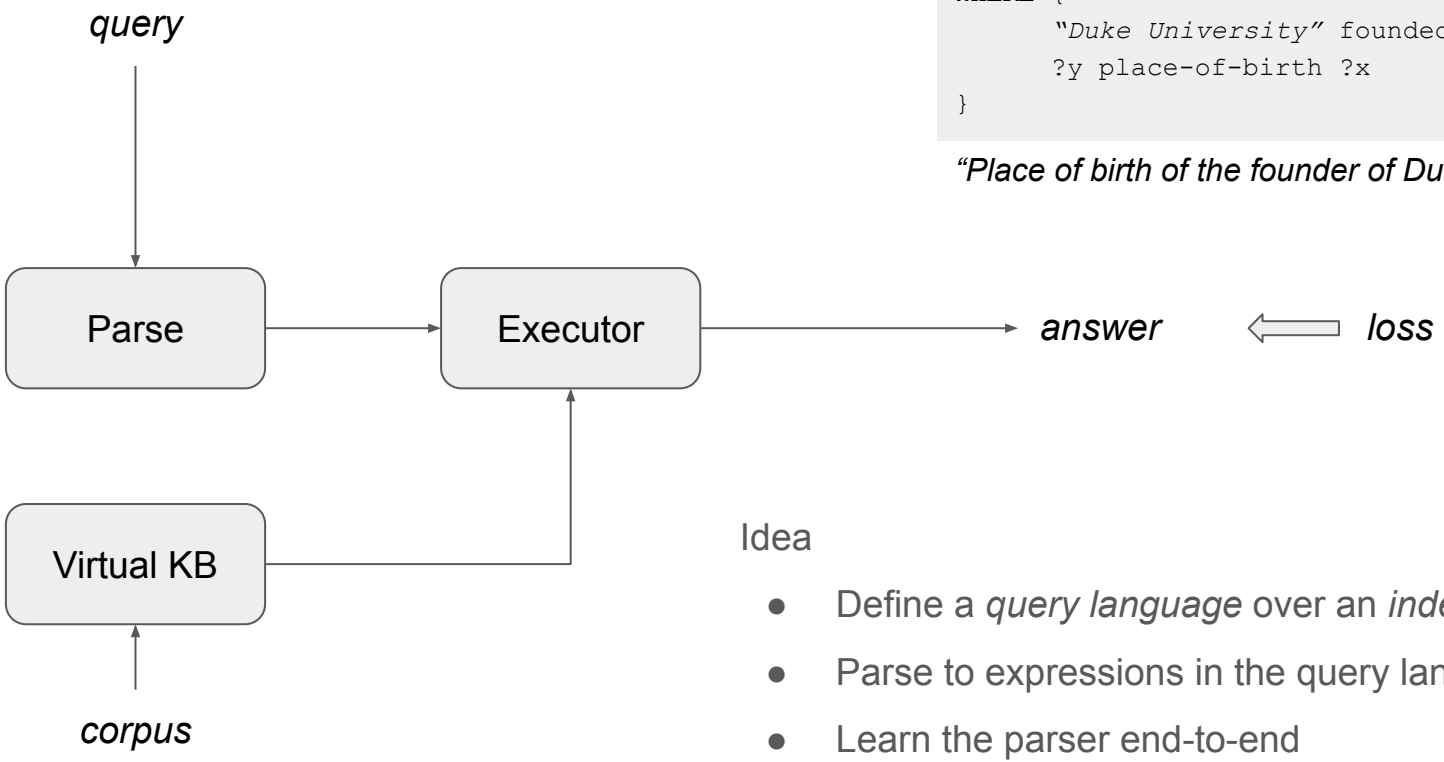
- For complex queries single-shot retrieval does not work

DrKIT

(Differentiable Reasoning over a KB of Indexed Text)

```
SELECT ?x
WHERE {
      "Duke University" founded-by ?y
      ?y place-of-birth ?x
}
```

"Place of birth of the founder of Duke University"

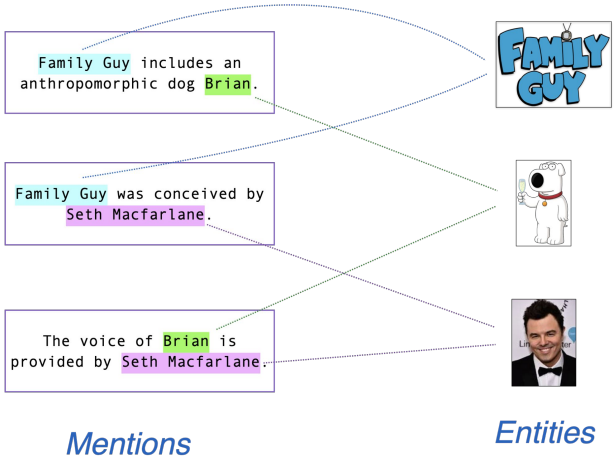


Idea

- Define a *query language* over an *indexed corpus*
- Parse to expressions in the query language
- Learn the parser end-to-end

Differentiable Reasoning over a Virtual Knowledge Base *Dhingra et al, 2020*

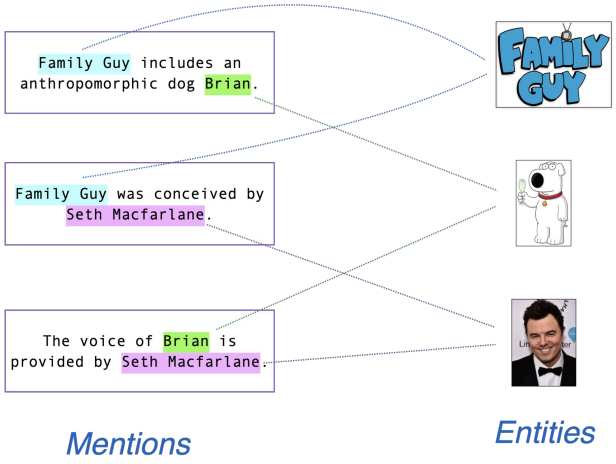
Virtual KB



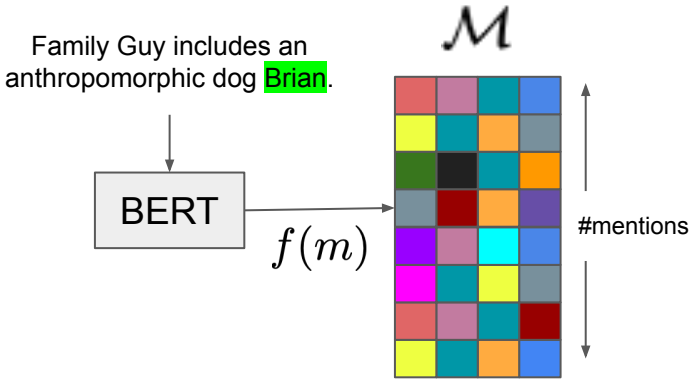
An entity-linked corpus

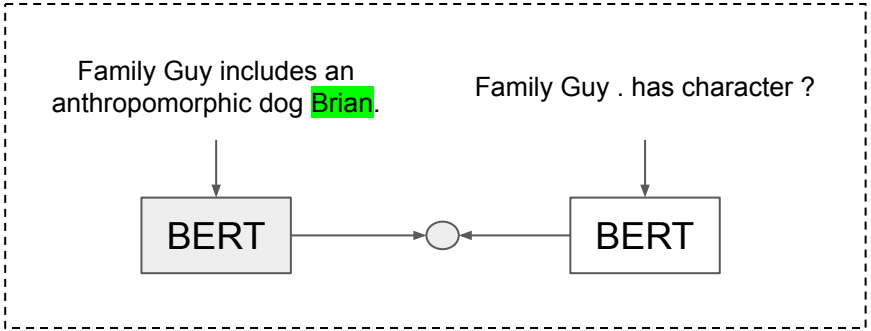
Virtual KB

1. Mention embeddings
   - Span start and end vectors from BERT
   - Pretrained on slot-filling over a KB



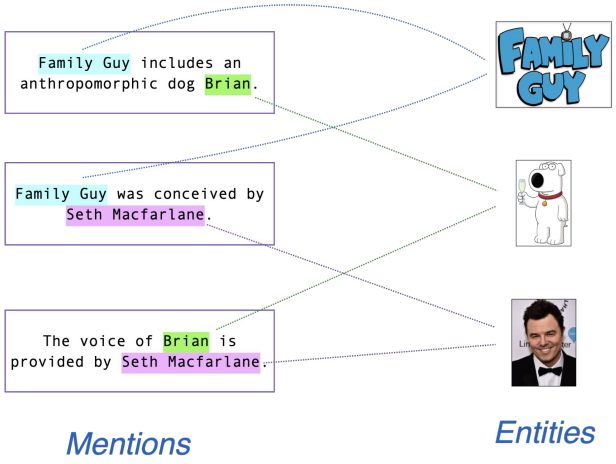
An entity-linked corpus



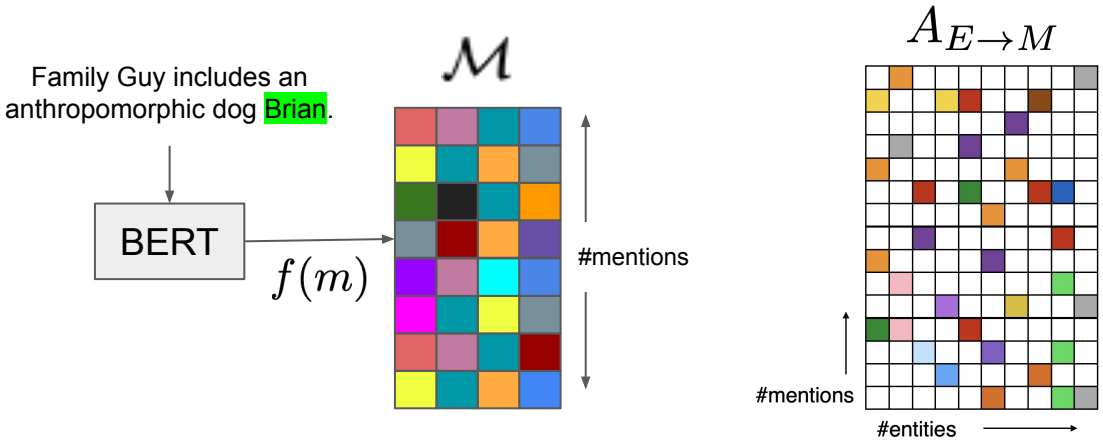


Virtual KB

1. Mention embeddings
2. Sparse co-occurrence matrix
   - All mentions which co-occur with an entity

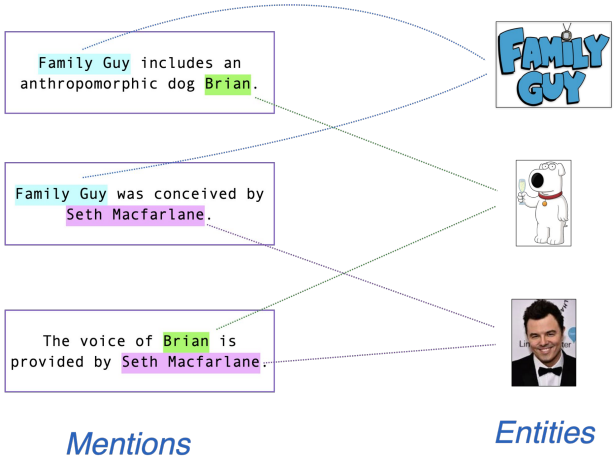


An entity-linked corpus

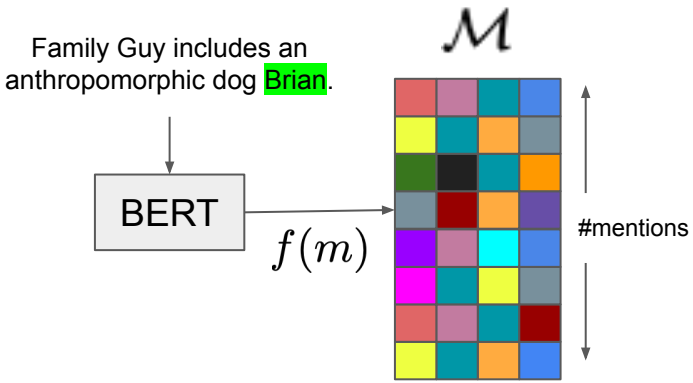


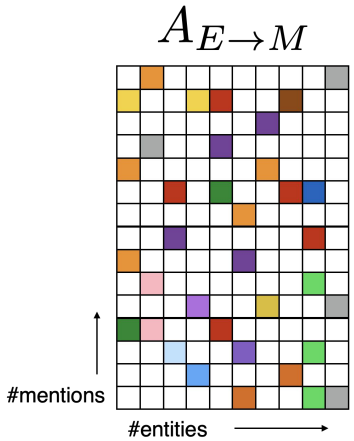
Virtual KB

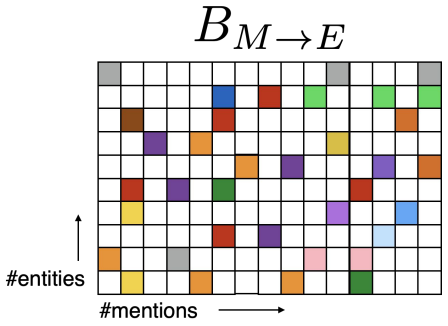
1. Mention embeddings
2. Sparse co-occurrence matrix
3. Sparse coreference matrix
   - All mentions of an entity



An entity-linked corpus

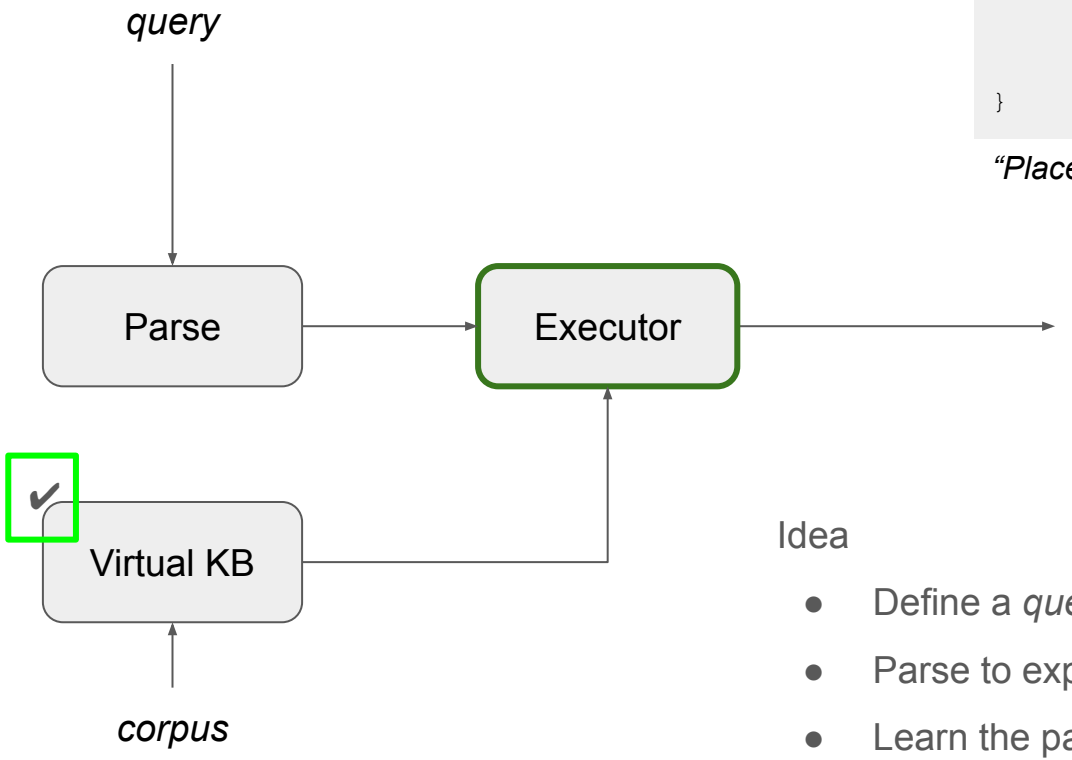






DrKIT

(Differentiable Reasoning over a KB of Indexed Text)




```
SELECT ?x
WHERE {
      "Duke University" founded-by ?y
      ?y place-of-birth ?x
}
```

"Place of birth of the founder of Duke University"

Idea

- Define a *query language* over an *indexed corpus*
- Parse to expressions in the query language
- Learn the parser end-to-end

Differentiable Reasoning over a Virtual Knowledge Base *Dhingra et al, 2020*

Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

- *X* and *Y* are sets of entities
- Useful for QA, e.g.,

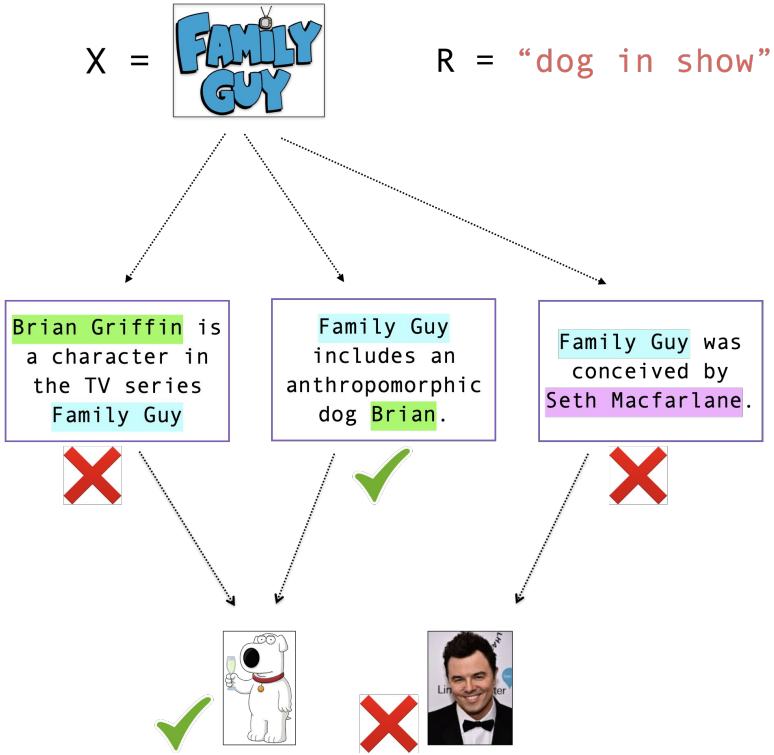
  "Who is the author of *On the Origin of Species*?"

  X = {*On the Origin of Species*}, R = {*author_of*} ⇒ Y = {*Charles Darwin*}

Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

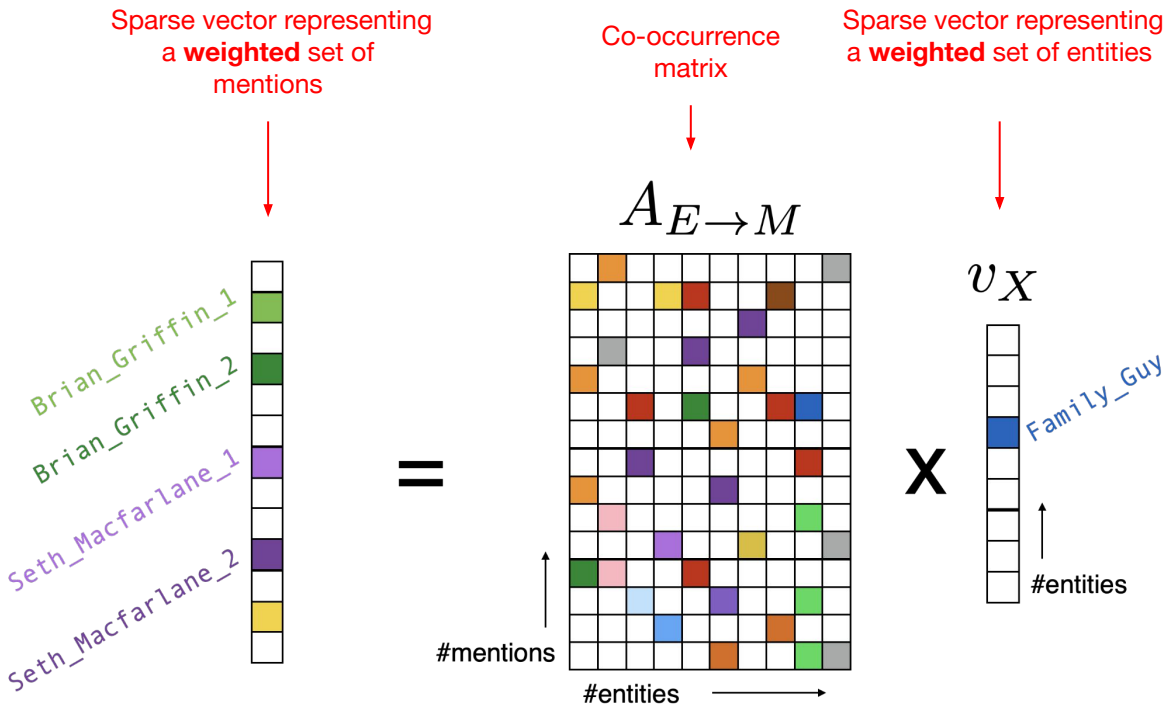
1. Expand *X* to co-occurring mentions
2. Filter mentions based on similarity to *R*
3. Aggregate over all mentions of the same entity



Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

1. Expand *X* to co-occurring mentions

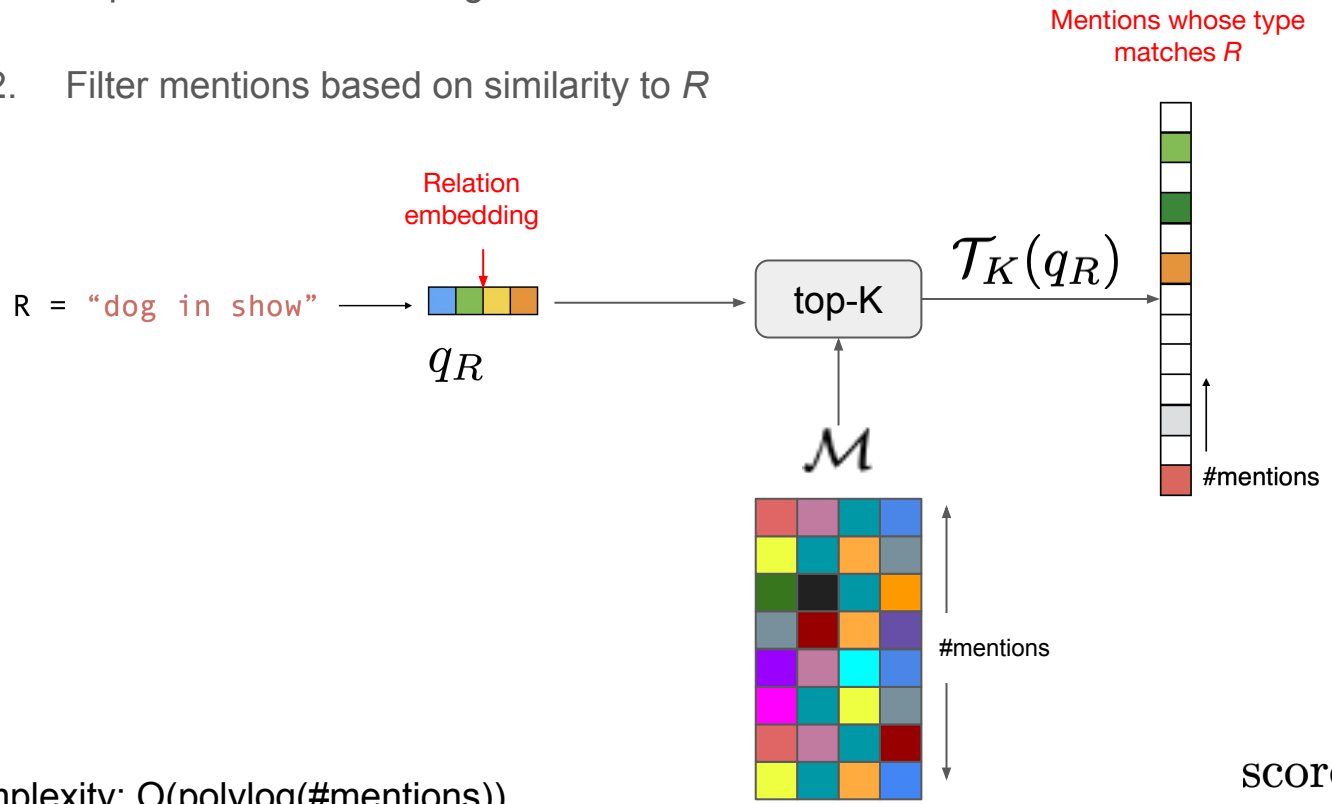


Complexity: O(|X| x out-degree)

Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

1. Expand *X* to co-occurring mentions
2. Filter mentions based on similarity to *R*



Mention embedding

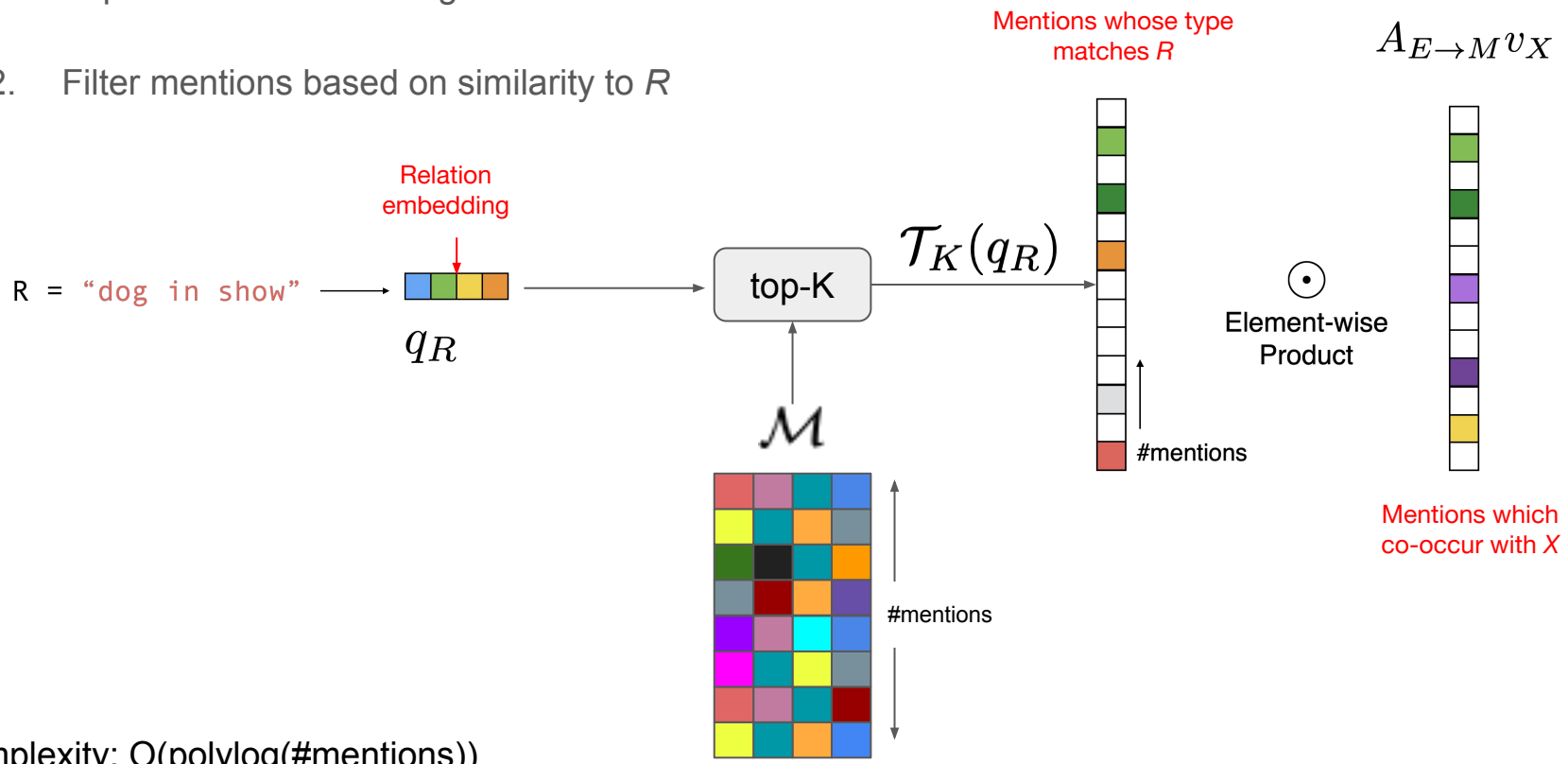
$$\text{score}(m, R) = f(m)^T q_R$$

Complexity: O(polylog(#mentions))

Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

1. Expand *X* to co-occurring mentions
2. Filter mentions based on similarity to *R*



Complexity: O(polylog(#mentions))

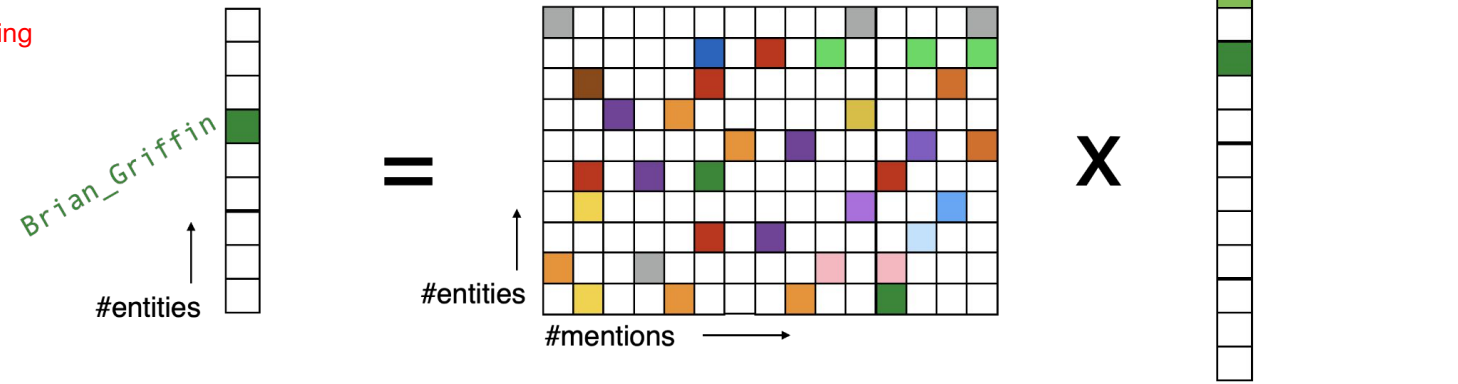
Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

1. Expand *X* to co-occurring mentions
2. Filter mentions based on similarity to *R*
3. Aggregate over all mentions of the same entity

$$v_{X.\text{follow}(R)} \quad B_{M \to E} \quad \mathcal{T}_K(q_R) \odot A_{E \to M} v_X$$

Sparse vector representing the **weighted** set of entities *Y*



Complexity: O(K)

Relation following

$$Y = X.\text{follow}(R) = \{x' \text{ s.t. } R(x, x') \text{ holds}\}$$

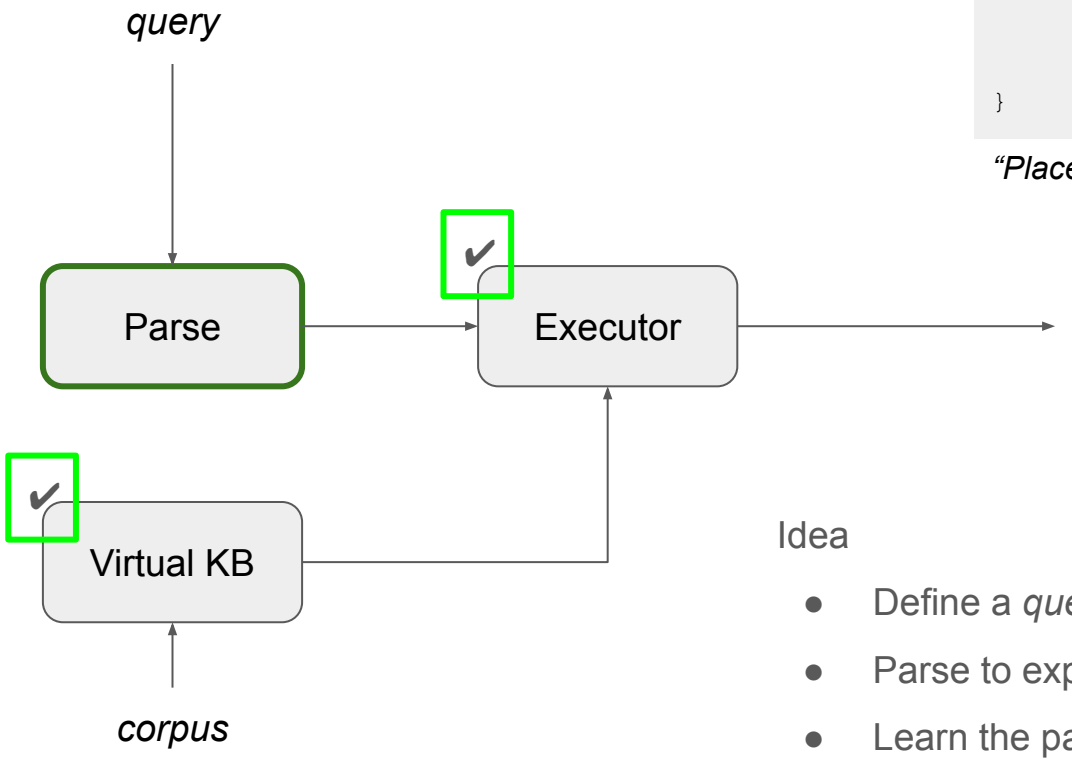
1. Expand *X* to co-occurring mentions
2. Filter mentions based on similarity to *R*
3. Aggregate over all mentions of the same entity

$$v_{X.\text{follow}(R)} = B_{M \to E}\left[\mathcal{T}_K(q_R) \odot A_{E \to M} v_X\right]$$

- ✓ Efficient
- ✓ Closed under composition
- ✓ Differentiable

DrKIT

(Differentiable Reasoning over a KB of Indexed Text)




```
SELECT ?x
WHERE {
      "Duke University" founded-by ?y
      ?y place-of-birth ?x
}
```

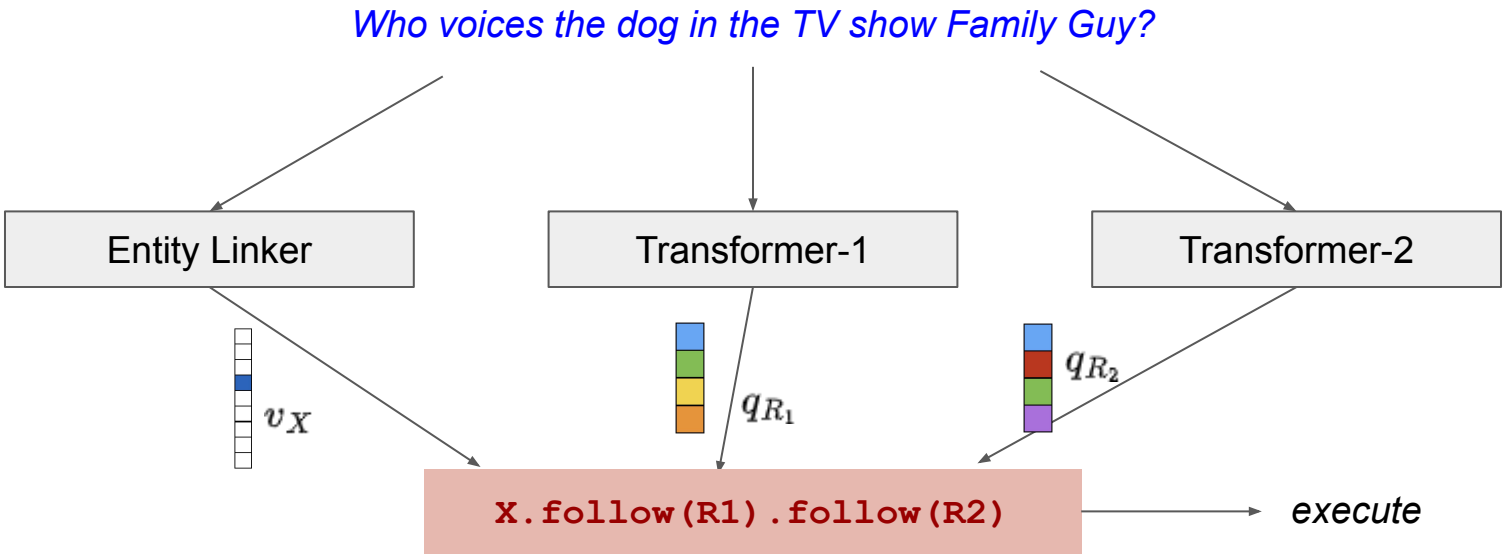
"Place of birth of the founder of Duke University"

Idea

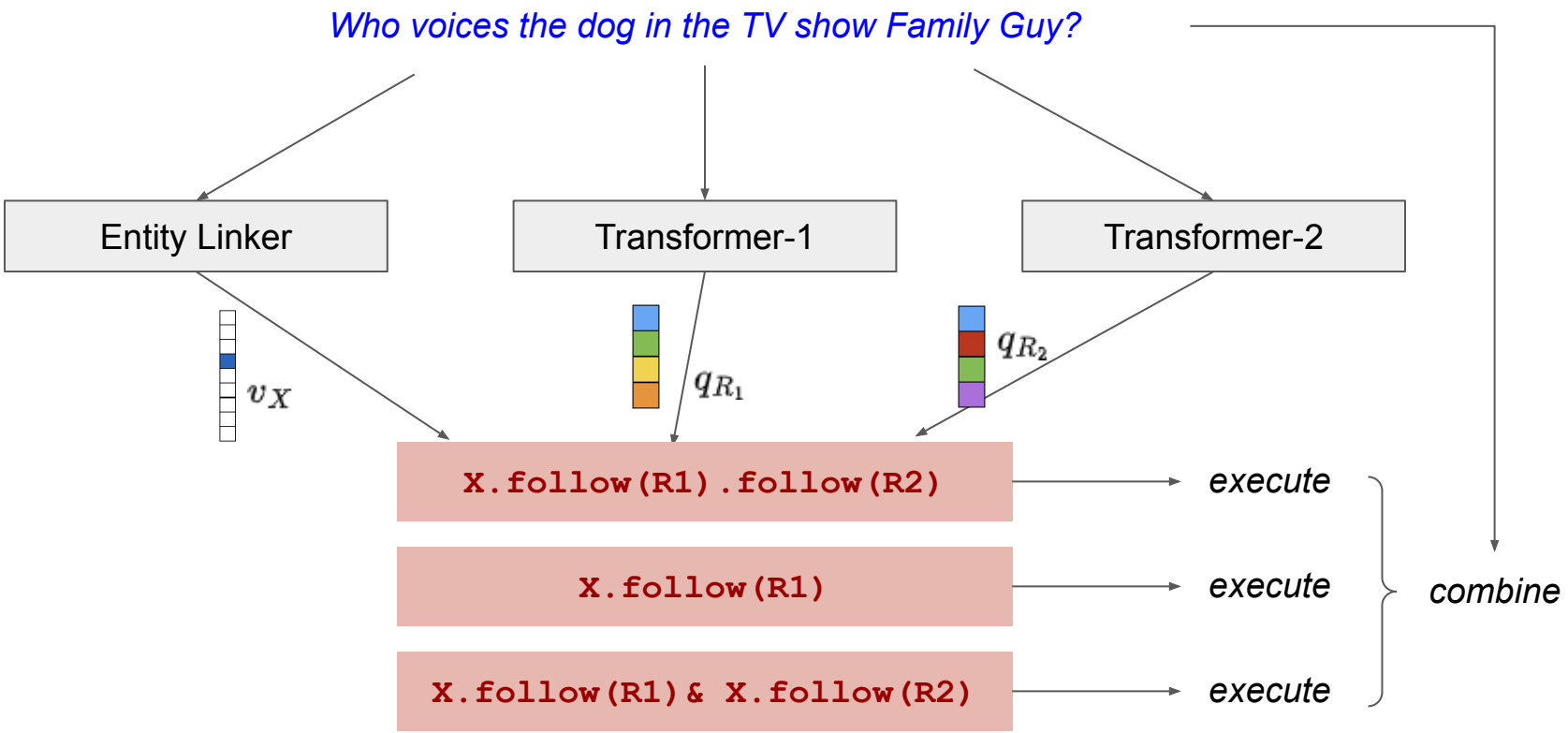
- Define a *query language* over an *indexed corpus*
- Parse to expressions in the query language
- Learn the parser end-to-end

Differentiable Reasoning over a Virtual Knowledge Base *Dhingra et al, 2020*

Query Templates



Mixing Templates



Results: Multi-Hop QA

	18K passages 43K entities 7 relations		120K passages 200K entities 888 relations	
	MetaQA		MSF	
Model	2Hop	3Hop	2Hop	3Hop
KVMem	7.0	19.5	3.4	2.6
DrQA	32.5	19.7	14.1	7.0
GRAFT-Net	36.2	40.2	-	-
PullNet	81.0	78.2	-	-
PIQA	-	-	36.9	18.2
DrKIT	86.0	**87.6**	46.9	24.4

Hits @1

Results: Multi-Hop QA

Retrieve once and then find answer

Model	MetaQA 2Hop	MetaQA 3Hop	MSF 2Hop	MSF 3Hop
KVMem	7.0	19.5	3.4	2.6
DrQA	32.5	19.7	14.1	7.0
GRAFT-Net	36.2	40.2	-	-
PullNet	81.0	78.2	-	-
PIQA	-	-	36.9	18.2
DrKIT	86.0	**87.6**	46.9	24.4

Hits @1

Results: Multi-Hop QA

Model	MetaQA 2Hop	MetaQA 3Hop	MSF 2Hop	MSF 3Hop
KVMem	7.0	19.5	3.4	2.6
DrQA	32.5	19.7	14.1	7.0
GRAFT-Net	36.2	40.2	-	-
PullNet	81.0	78.2	-	-
PIQA	-	-	36.9	18.2
DrKIT	86.0	**87.6**	46.9	24.4

Retrieve iteratively but not end-to-end

Hits @1

Results: Multi-Hop QA

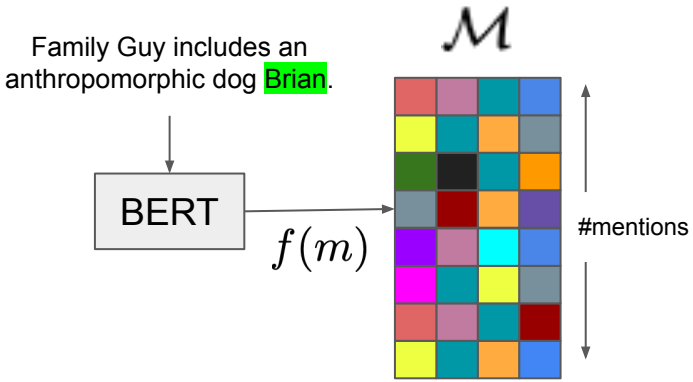
Model	MetaQA 2Hop	MetaQA 3Hop	MSF 2Hop	MSF 3Hop
KVMem	7.0	19.5	3.4	2.6
DrQA	32.5	19.7	14.1	7.0
GRAFT-Net	36.2	40.2	-	-
PullNet	81.0	78.2	-	-
PIQA	-	-	36.9	18.2
DrKIT	86.0	**87.6**	46.9	24.4

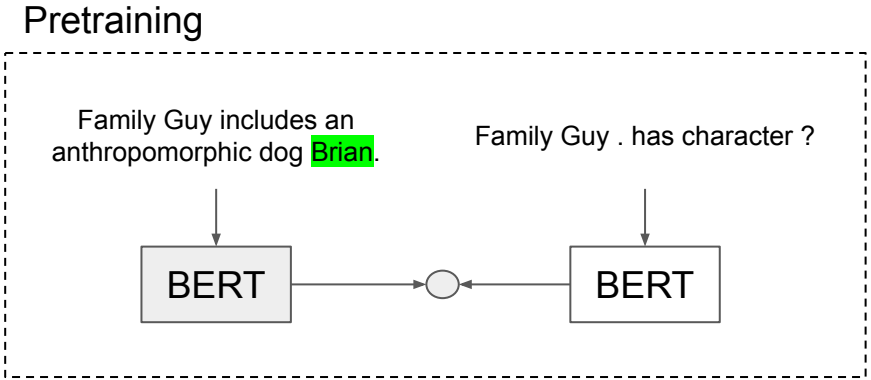
3x-15x faster than baselines

Hits @1

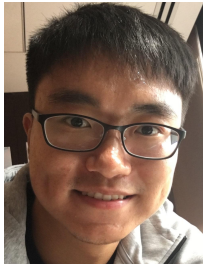
Limitations

- Pretraining the mention embeddings requires an existing KB
  - Not available in every domain
  - Lower accuracy on relations not in the KB
- Same mention participates in different relations



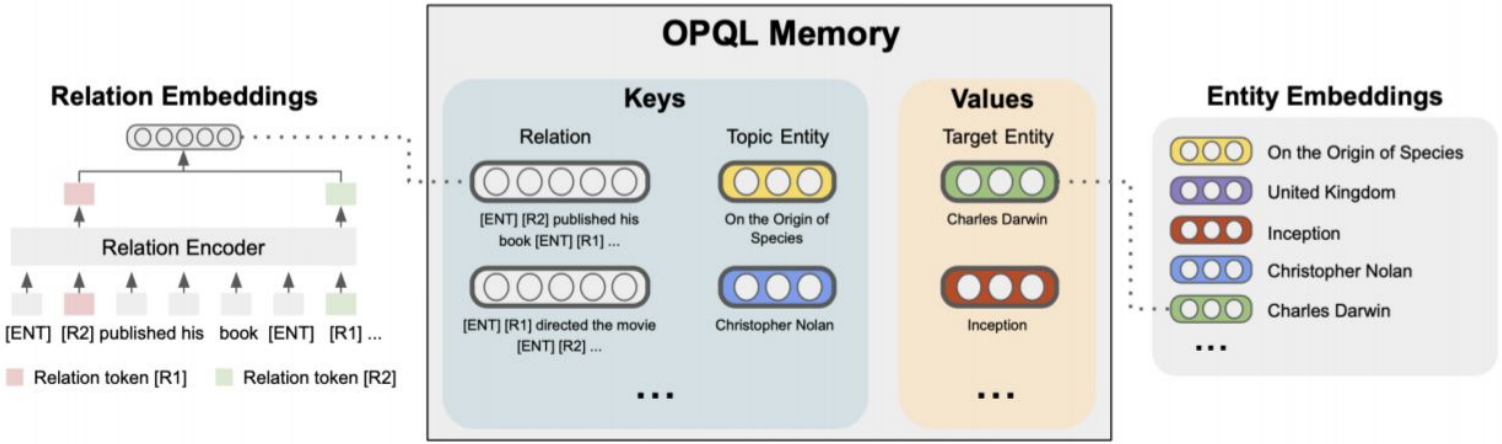


OPQL: Open Predicate Query Language



Haitian Sun

Virtual KB is a *key-value memory* over *pairs of entity mentions*



***Charles Darwin** published his book **Origin of the Species** after waiting ….*

Reasoning Over Virtual Knowledge Bases With Open Predicate Relations *Sun et al, 2021*

OPQL: Open Predicate Query Language

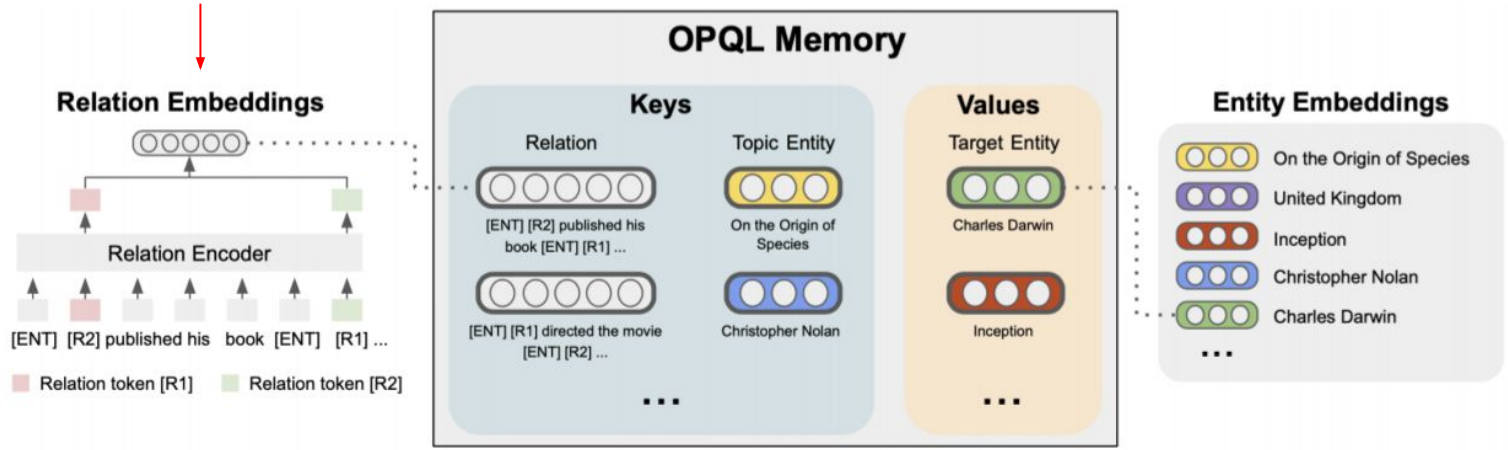
Relation embedding

$$\mathbf{q}_{X,Y} = \mathbf{W}_q^T [\mathbf{e}_X ; \mathbf{W}_t^T \mathbf{r}_{X,Y}]$$

Query for top-K search

Entity embedding (from previous hop)

We can pretrain this without a KB by matching entity pair mentions!



Charles Darwin *published his book* ***Origin of the Species*** *after waiting ….*

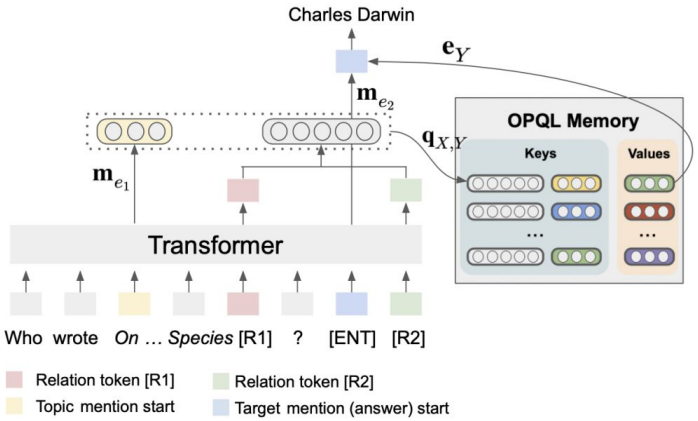
Matching the Blanks: Distributional Similarity for Relation Learning *Soares et al, 2019*

Results: Multi-Hop QA

Model	MetaQA 2Hop	MetaQA 3Hop	MSF 2Hop	MSF 3Hop
KVMem	7.0	19.5	3.4	2.6
DrQA	32.5	19.7	14.1	7.0
GRAFT-Net	36.2	40.2	-	-
PullNet	81.0	78.2	-	-
PIQA	-	-	36.9	18.2
DrKIT	86.0	**87.6**	46.9	24.4
OPQL-pretrained	84.7	84.3	48.5	28.1
OPQL	**88.5**	87.1	**49.2**	**29.7**

Relation encoder is tuned on domain specific relations

More results: Open-domain QA



Model	WebQSP	ComplexWebQ (dev)
GRAFT-NET	25.3	10.6
PullNet	24.8	13.1
BART-Large	30.4	-
EaE	47.4	31.3
DPR	48.6	24.6
DPR-cascade	-	25.1
T5	49.7	38.7
OPQL-follow	46.6	18.5
OPQL-LM	**51.9**	**40.7**

Mixing relation following results with a language model

Summary

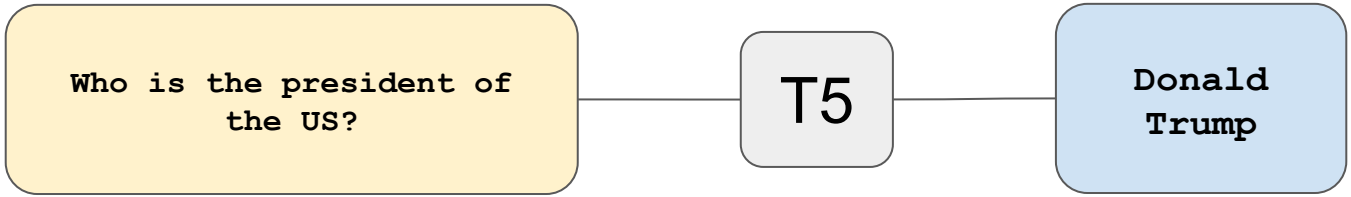
- How can we perform explicit reasoning in a neural network?
  - By defining differentiable query languages over preprocessed corpora
  - Much related work over structured KBs (NQL, EmQL, Query2Box)
- How can we make the query language more expressive?
  - Conjunctions and disjunctions are possible but not well tested
  - Numerical operations are more tricky

Overview

1. Differentiable query language over text ✔
   - ➢ Differentiable Reasoning over a Virtual KB *Dhingra et al, ICLR 2020*
   - ➢ Reasoning Over Virtual KBs With Open Predicate Relations *Sun et al, ICML 2021*

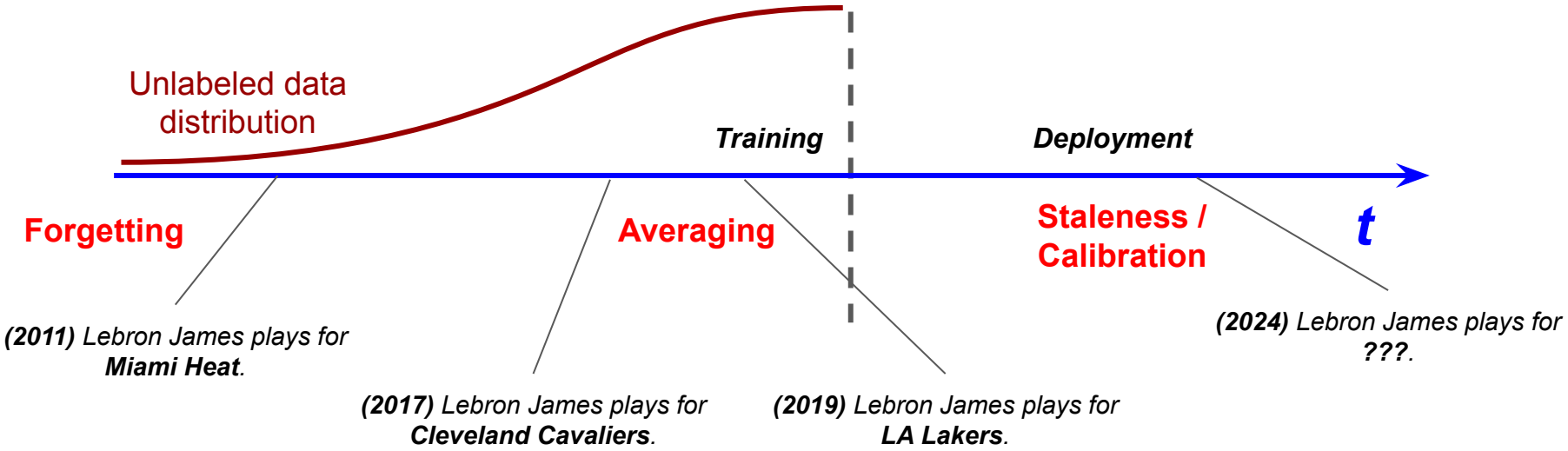
2. Adding temporal scopes to pretrained knowledge inside LMs
   - ➢ Time-Aware LMs as Temporal KBs *Dhingra et al, 2021 (Under Review)*

Knowledge changes with time



- Do LMs learn the temporal scope of the facts they encode?
- How can we update temporally-scoped knowledge in trained models?

Training Data Timeline



TempLAMA: A diagnostic dataset

- Identify Wikidata facts with multiple objects across time

- Convert to masked LM queries

  t = 2012

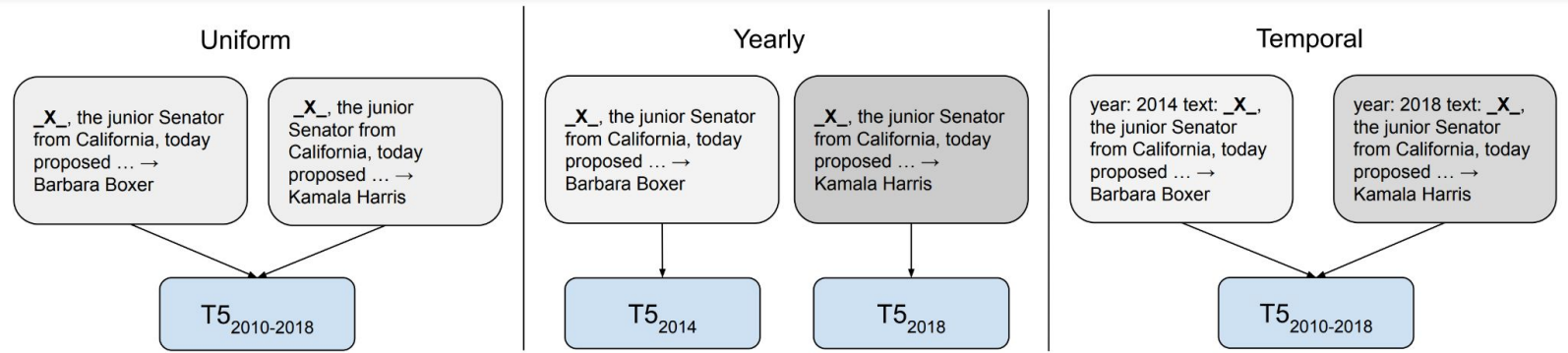
  x = "Lebron James plays for ___"

  y = "Miami Heat"

Time-Aware Language Models as Temporal Knowledge Bases *Dhingra et al, 2021*

Time-aware pretraining

- Instead of **Pr(y|x)** model **Pr(y|x, t)**



- Pretraining data: CustomNews (Lazaridou et al, 2020)
  - 1M news articles each from 2010-2018
  - Mask out salient spans (entities)

- Start with T5 pretrained for 1M steps on C4 (April 2019 crawl)

Results

T5 closed-book QA model struggles on time-sensitive questions

Model	#Parameters	TempLAMA		
		2010-18	2019-20	Overall
T5-CBQA	737M	5.4	4.3	5.2
T5-CBQA-ft	737M	17.8	15.3	17.3
Uniform	737M	28.1	19.8	26.6
Yearly	6.6B	28.5	21.8	27.3
Temporal	737M	**29.6**	**22.2**	**28.2**

Token-level F1

All models are trained on data prior to 2019

Results

Pretraining on uniformly sampled news helps

(~4% inputs mention a date)

Model	#Parameters	TempLAMA		
		2010-18	2019-20	Overall
T5-CBQA	737M	5.4	4.3	5.2
T5-CBQA-ft	737M	17.8	15.3	17.3
Uniform	737M	28.1	19.8	26.6
Yearly	6.6B	28.5	21.8	27.3
Temporal	737M	**29.6**	**22.2**	**28.2**

Token-level F1

All models are trained on data prior to 2019

Results

Single model with time prefixes does better than ensemble

Model	#Parameters	TempLAMA		
		2010-18	2019-20	Overall
T5-CBQA	737M	5.4	4.3	5.2
T5-CBQA-ft	737M	17.8	15.3	17.3
Uniform	737M	28.1	19.8	26.6
Yearly	6.6B	28.5	21.8	27.3
Temporal	737M	**29.6**	**22.2**	**28.2**

Token-level F1

All models are trained on data prior to 2019

Results

Model	#Parameters	TempLAMA		
		2010-18	2019-20	Overall
T5-CBQA	737M	5.4	4.3	5.2
T5-CBQA-ft	737M	17.8	15.3	17.3
Uniform	737M	28.1	19.8	26.6
Yearly	6.6B	28.5	21.8	27.3
Temporal	737M	**29.6**	**22.2**	**28.2**

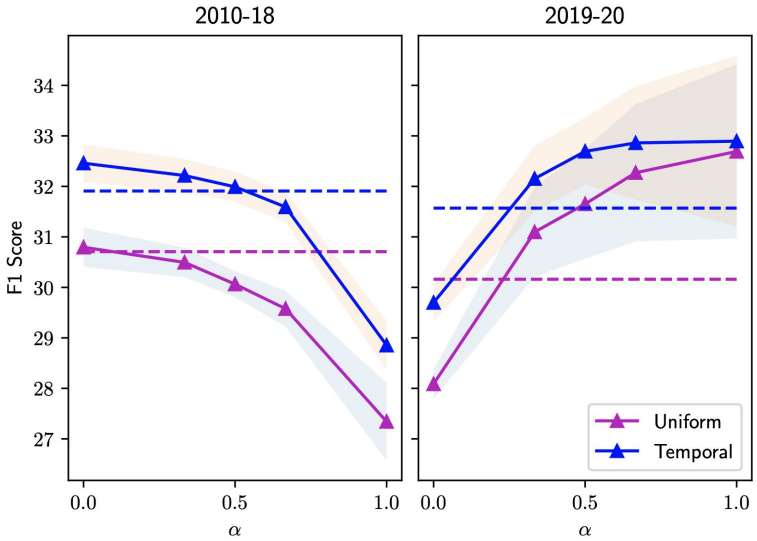
Token-level F1

On future slice models only get unchanged facts correct

All models are trained on data prior to 2019

Updates without forgetting

- Setup: finetune models trained on 2010-28 on new data from 2019
- To avoid forgetting mix old and new data and train for 50K steps
- Temporal prefixes lead to less forgetting



Fraction of new data when finetuning

Summary

- Time-aware pretraining helps
  - Organize temporal knowledge inside an LM
  - Add new knowledge to the LM
- What is the best way of modeling time in LMs?
  - String prefixes are easy but don’t have any inductive bias about the continuity of time
  - Lots of related work on temporal knowledge graphs (e.g. HyTE; Dasgupta et al, 2018)
- What other metadata can be useful?

Thank you.