Differential Reasoning Over a Virtual Knowledge Base

Authors: Bhuwan Dhingra , Manzil Zaheer , Vidhisha Balachandran , Graham Neubig , Ruslan Salakhutdinov , William W. Cohen

Presented By: Saransh Goyal

Multi-Hop Question Answering over Virtual KBs

- Text as Virtual Knowledge Bases
  - More complete
  - Errors from extraction process don’t propagate
  - Answers are spans from text
- Multi-Hop
  - 1,2,3-Hop Questions
  - Can be answered from both KB and text

Motivation

- QA models too expensive
- Multi-hop not easy for them
- Differential versions of KB operations over text

Overall Idea

- Treat text as KBs
- Neural/probabilistic versions of KB operations
  - Retrieve entities
  - Perform hops on entities
- Many optimisations to make an efficient pipeline

Pipeline

- Get entities(z) from the question (q)
- Get co-occurring mentions(m) for z
  - Using TF-IDF for co-occurence
  - Neural network to filter relevant(to q) mentions
- Aggregate over m to get z’ candidate entities
- Multi-hop
  - Start with z’ entities

Getting relevant entities and mentions

$$\Pr(z_t|q) = \sum_{m \in \mathcal{M}} \sum_{z_{t-1} \in \mathcal{E}} \Pr(z_t|m) \Pr(m|q, z_{t-1}) \Pr(z_{t-1}|q)$$

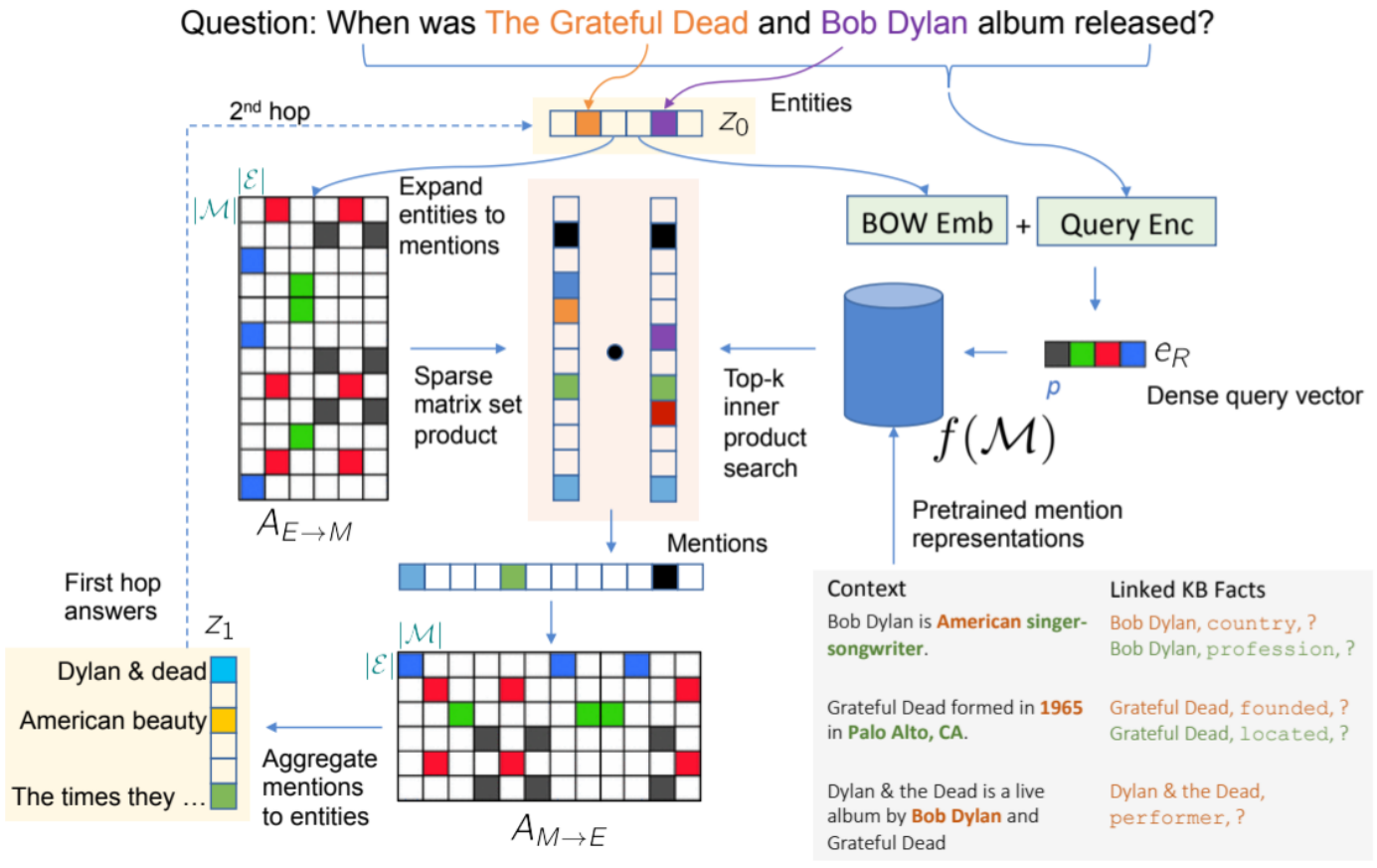
$$\Pr(m|q, z_{t-1}) \propto \underbrace{\mathbb{1}\{G(z_{t-1}) \cdot F(m) > \epsilon\}}_{\text{expansion to co-occurring mentions}} \times \underbrace{s_t(m, z_{t-1}, q)}_{\text{relevance filtering}}$$

F(m): TF-IDF of document containing m

G(z_{t-1}): TF-IDF of surface form of entity from previous step

Equations for a step

- $A_{E \to M}[e, m] = 1\,(G(e) \cdot F(m) > \epsilon)$
- Sparse matrix-vector multiplication
  - $A_{E \to M}$ and z_{t-1}
- $T_K\,(s_t\,(m, z_{t-1}, q))$: top-K relevant mentions
- $A_{M \to E}$: Encode co-reference
- $Z_t = \text{softmax}([Z^T_{t-1} A_{E \to M} \odot T_K(s_t(m, z_{t-1}, q))]\, A_{M \to E})$


Question: When was The Grateful Dead and Bob Dylan album released?
2nd hop
Entities
z_0
$|\mathcal{E}|$
$|\mathcal{M}|$
Expand entities to mentions
$A_{E \to M}$
Sparse matrix set product
Mentions
BOW Emb
+
Query Enc
e_R
Dense query vector
p
Top-k inner product search
$f(\mathcal{M})$
Pretrained mention representations
Context
Linked KB Facts
Bob Dylan is American singer-songwriter.
Bob Dylan, country, ?
Bob Dylan, profession, ?
Grateful Dead formed in 1965 in Palo Alto, CA.
Grateful Dead, founded, ?
Grateful Dead, located, ?
Dylan & the Dead is a live album by Bob Dylan and Grateful Dead
Dylan & the Dead, performer, ?
First hop answers
z_1
Dylan & dead
American beauty
The times they …
Aggregate mentions to entities
$A_{M \to E}$

Implementation Optimisations

Sparse TF-IDF Mention Encoding

- TF-IDF over uni,bi-grams, hashed to 16M buckets
- Max μ mentions per entity

Efficient Entity-Mention expansion

- 2 row-wise list of lists of non-zero values and indexes
- Feasible with TensorFlow RaggedTensors

Implementation Optimisations Contd.

Efficient top-k mention relevance filtering

- Restrict $s_t(m, z_{t-1}, q)$ to an inner product
- $f(m)$ be a dense encoding of m
- $g_t(q, z_{t-1})$ be a dense encoding of the question q for the t-th hop
- $s_t(m, z_{t-1}, q) \propto \exp\{f(m) \cdot g_t(q, z_{t-1})\}$
- Use approximate algorithm for Maximum Inner Product Search (MIPS)

Mention and Question Encoders

- Passage containing mention is passed through Bert-Large
- $f(m) = W^T [H^d_i ; H^d_j]$, H^d is the output embeddings from Bert
- Queries are passed through a 4-layer transformer, outputting H^q
- Add 2 transformer layers on H^q to output start and end tokens H^q_{st} and H^q_{en}
- $g_t(q, z_{t-1}) \equiv V^T [H^q_{st} ; H^q_{en}] + Z^T_{t-1}$

Pre-training the Index

- Pre-train f(m) and then keep it fixed during QA task
- Facts given in form (e1, R, e2)
- Find passages, d which contain both e1 and e2
- Convert to slot-filling task
- Negative instances: shared entity(e2 missing), shared relation(e1’,e2’), random
- WikiData as KB, Wikipedia as corpus

Experiments

- MetaQA: Multi-Hop QA
- WikiData: Multi-Hop Slot-Filling
  - e.g. “Helene Gayle, employer, founded by, ?”
- HotpotQA: Multi-Hop Information Retrieval
  - Retrieve relevant 2 documents, then use their baseline MRC model

Experiments Results

MetaQA			
Model	**1hop**	**2hop**	**3hop**
DrQA (ots)	0.553	0.325	0.197
KVMem†	0.762	0.070	0.195
GraftNet†	0.825	0.362	0.402
PullNet†	0.844	0.810	0.782
DrKIT (e2e)	0.844	0.860	**0.876**
DrKIT (strong sup.)	**0.845**	**0.871**	0.871

WikiData			
Model	**1hop**	**2hop**	**3hop**
DrQA (ots, cascade)	0.287	0.141	0.070
PIQA (ots, cascade)	0.240	0.118	0.064
PIQA (pre, cascade)	0.670	0.369	0.182
DrKIT (pre, cascade)	0.816	0.404	0.198
DrKIT (e2e)	**0.834**	**0.469**	**0.244**
–BERT index	0.643	0.294	0.165

Table 1: **(Left)** MetaQA and **(Right)** WikiData Hits @1 for 1-3 hop sub-tasks. ots: off-the-shelf without re-training. †: obtained from Sun et al. (2019). cascade: adapted to multi-hop setting by repeatedly applying Eq. 2. pre: pre-trained on slot-filling. e2e: end-to-end trained on single-hop and multi-hop queries.

PullNet: Graph Neural Network based, uses MetaQA KB for strong intermediate supervision

Ablation Study

Ablations	**1hop**	**2hop**	**3hop**
DrKIT	0.844	0.86	0.876
–Sum over M_{z_t}	0.837	0.823	0.797
$-\lambda = 1$	0.836	0.752	0.799
–w/o TFIDF	0.845	0.548	0.488
–BERT index	0.634	0.610	0.555
Incomplete KB for pretraining			
25% KB	0.839	0.804	0.830
50% KB	0.843	0.834	0.834
(50% KB-only)	*0.680*	*0.521*	*0.597*

HotpotQA Results

Model	Q/s	Accuracy			
		@2	@5	@10	@20
BM25†	–	0.093	0.191	0.259	0.324
PRF-Task†	–	0.097	0.198	0.267	0.330
BERT re-ranker†	–	0.146	0.271	0.347	0.409
Entity Centric IR†	0.32*	0.230	0.482	0.612	0.674
DrKIT (WikiData)	**4.26***	0.355	0.588	0.671	**0.710**
DrKIT (Hotpot)		**0.385**	0.595	0.663	0.703
DrKIT (Combined)		0.383	**0.603**	**0.672**	**0.710**

Model	EM	F1
Baseline†	0.288	0.381
+EC IR‡	0.354	0.462
+Golden Ret◇	**0.379**	**0.486**
+DrKIT†	0.357	0.466

Table 2: **(Left)** Retrieval performance on the HotpotQA benchmark dev set. Q/s denotes the number of queries per second during inference on a single 16-core CPU. Accuracy @k is the fraction where *both* the correct passages are retrieved in the top k. †: Baselines obtained from Das et al. (2019b). For DrKIT, we report the performance when the index is pretrained using the WikiData KB alone, the HotpotQA training questions alone, or using both. *: Measured on different machines with similar specs. **(Right)** Overall performance on the HotpotQA task, when passing 10 retrieved passages to a downstream reading comprehension model (Yang et al., 2018). ‡: From Das et al. (2019b). ◇: From Qi et al. (2019). †: Results on the dev set.

Pros

- Mathematical Explanation (Atishya)
- Work on scalability (Atishya, Jigyasa)
- Ablation Study (Atishya, Pratyush)
- Speedup tricks (Jigyasa, Pratyush)
- Avoids noise from IE step(Pratyush, Pawan)
- SOTA (Pratyush, Shubham)
- Virtual KB using mentions (Pawan)
- End-to-end (Shubham)
- MIPS (Shubham)
- Bridge gap between KB and text corpus (Shubham)

Cons

- Predefined entities (Atishya)
- Number of hops decision (Atishya, Jigyasa, Shubham)
- Max vs sum (Atishya)
- Constraint on types of questions due to pre-training (Jigyasa)
- Lots of clever engineering (Jigyasa)
- Static Index (Shubham)
- Entity linked corpus needed (Shubham)
- Limited to entity spans (Pratyush)
- Not really e2e due to pretraining (Pratyush)
- More detail needed on mention encoder (Pawan)

Extensions

- Fine-tune mention encoder(Atishya)
- Module to estimate number of hops (Atishya)
- Use of KB at test time (Jigyasa)
- Use of Word2vec instead of TF-IDF (Jigyasa)
- Weighted sum over mentions (Pratyush)
- Symbolic reasoning over mentions over hops (Pawan)