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-occurance
  - 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

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

\[ \Pr(m|q, z_{t-1}) \propto \mathbb{1}\{G(z_{t-1}) \cdot F(m) > \epsilon\} \times s_t(m, z_{t-1}, q) \]

\begin{itemize}
  \item \text{F(m): TF-IDF of document containing m}
  \item \text{G(z_{t-1}): TF-IDF of surface form of entity from previous step}
\end{itemize}
Equations for a step

- \( A_{E\rightarrow M}[e, m] = 1 \) (G(e) \cdot F(m) > \epsilon)
- Sparse matrix-vector multiplication
  - \( A_{E\rightarrow M} \) and \( z_{t-1} \)
- \( T_K(s_t(m, z_{t-1}, q)) \): top-K relevant mentions
- \( A_{M\rightarrow E} \): Encode co-reference
- \( Z_t = \text{softmax}([Z_{t-1}^T A_{E\rightarrow M} \odot T_K(s_t(m, z_{t-1}, q))] A_{M\rightarrow E}) \)
Question: When was The Grateful Dead and Bob Dylan album released?
Implementation Optimisations

Sparse TF-IDF Mention Encoding

- TF-IDF over uni,bi-grams, hashed to 16M buckets
- Max $\mu$ 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 \((e2 \text{ missing})\), 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

<table>
<thead>
<tr>
<th>Model</th>
<th>MetaQA</th>
<th>WikiData</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1hop</td>
<td>2hop</td>
</tr>
<tr>
<td>DrQA (ots)</td>
<td>0.553</td>
<td>0.325</td>
</tr>
<tr>
<td>KVMem†</td>
<td>0.762</td>
<td>0.070</td>
</tr>
<tr>
<td>GraftNet†</td>
<td>0.825</td>
<td>0.362</td>
</tr>
<tr>
<td>PullNet†</td>
<td>0.844</td>
<td>0.810</td>
</tr>
<tr>
<td>DrKIT (e2e)</td>
<td>0.844</td>
<td>0.860</td>
</tr>
<tr>
<td>DrKIT (strong sup.)</td>
<td>0.845</td>
<td>0.871</td>
</tr>
<tr>
<td>DrQA (ots, cascade)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIQA (ots, cascade)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIQA (pre, cascade)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DrKIT (pre, cascade)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DrKIT (e2e) -BERT index</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: (Left) MetaQA and (Right) WikiData Hits @1 for 1-3 hop sub-tasks. ots: off-the-shelf without retraining. †: 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

<table>
<thead>
<tr>
<th>Ablations</th>
<th>1hop</th>
<th>2hop</th>
<th>3hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrKIT</td>
<td>0.844</td>
<td>0.86</td>
<td>0.876</td>
</tr>
<tr>
<td>– Sum over $M_{zt}$</td>
<td>0.837</td>
<td>0.823</td>
<td>0.797</td>
</tr>
<tr>
<td>– $\lambda = 1$</td>
<td>0.836</td>
<td>0.752</td>
<td>0.799</td>
</tr>
<tr>
<td>– w/o TFIDF</td>
<td>0.845</td>
<td>0.548</td>
<td>0.488</td>
</tr>
<tr>
<td>– BERT index</td>
<td>0.634</td>
<td>0.610</td>
<td>0.555</td>
</tr>
</tbody>
</table>

*Incomplete KB for pretraining*

<table>
<thead>
<tr>
<th>KB percentage</th>
<th>1hop</th>
<th>2hop</th>
<th>3hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>25% KB</td>
<td>0.839</td>
<td>0.804</td>
<td>0.830</td>
</tr>
<tr>
<td>50% KB</td>
<td>0.843</td>
<td>0.834</td>
<td>0.834</td>
</tr>
<tr>
<td>(50% KB-only)</td>
<td>0.680</td>
<td>0.521</td>
<td>0.597</td>
</tr>
</tbody>
</table>
## HotpotQA Results

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

<table>
<thead>
<tr>
<th>Model</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline‡</td>
<td>0.288</td>
<td>0.381</td>
</tr>
<tr>
<td>+EC IR‡</td>
<td>0.354</td>
<td>0.462</td>
</tr>
<tr>
<td>+Golden Ret(\odot)</td>
<td>0.379</td>
<td>0.486</td>
</tr>
<tr>
<td>+DrKIT‡</td>
<td>0.357</td>
<td>0.466</td>
</tr>
</tbody>
</table>

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). \(\odot\): 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)