Scribe Note: KG^2: Learning to Reason Science Exam Questions with Contextual Knowledge Graph Embeddings[2]

Presenter: William Zhang, Scribe: Ji Gao

June 1, 2019

1 Task

- Answer science exam questions
- The science exam is targeting 8-13 year old student, however, previous IR methods can only get 21% accuracy, even worse than random guess.
- Motivation: Use human-like logic to help solving the questions:
  1. Read the question
  2. Generate hypothesis by combining the question stem and answer option
  3. Find supporting sentences in the corpus
  4. Verify the hypothesis
- Use knowledge graph and deep neural models to help the model.

2 Method

- Definitions:
  - i-th Question stem(text): q_i, where $i \in \{1, 2...n\}$
  - Answer(text) $c_i^{(j)}$, where $i \in \{1, 2...n\} \wedge j \in \{1, 2..m\}$
  - Science exam questions: $\mathcal{D} = \left\{q_i,\ \left(c_i^{(1)}, \ldots, c_i^{(m)}\right),\ a_i\right\}_{i=1}^{n}$, a_i is the label of correct answer.
- Generating Hypothesis:
  - If wh-word can be found in the question, replacing the wh-word with the answer.

- If no wh-word can be found, append the correct answer to the question.

- Searching Potential Supports:
  - Query the hypothesis in the corpus(Which is very large)
  - Use ElasticSearch to pick top 20 sentences

- Constructing Knowledge Graphs
  - Use Open IE[1] to generate a knowledge graph.
  - Extract relation triple $T(s, p, o_i)$, s is the subject, p is predicate, and o_i is i-th object. In the graph, it will build edge *subj* and *obj* between s and o

- Inference with graph embedding
  - Evaluates $f : G_{hypo} \times G_{supp} \rightarrow \mathcal{R}$ on every q, c pair, and pick the best c
  - Use a graph model to evaluate. Iteratively, for every v, calculate an embedding

$$\mu_v^{(t)} = h\Big(\mathbf{x}_v, \mu_v^{(t-1)}, \{(\mu_u^{(t-1)}, e_{u,v})\}_{(u,v,e_{u,v})\in E}\Big) \tag{1}$$

    Where $\mathbf{x}_v$ encodes the text feature.
    $e_{u,v}$ stands for the edge type, that can be *time*, *loc*, etc.
  - After T iterations, the model returns the maximum cosine similarity result.

$$\begin{aligned} f(G^{hypo}, G^{supp}) &= f(\{\mu_u\}_{u\in V_p^{hypo}}, \{\mu_v\}_{v\in V_p^{supp}}) \\ &= \sigma\Big(\max_{u,v} \frac{\mu_u^\top \mu_v}{\|\mu_u\|\|\mu_v\|} - 0.5\Big), \end{aligned} \tag{2}$$

3 Experiment

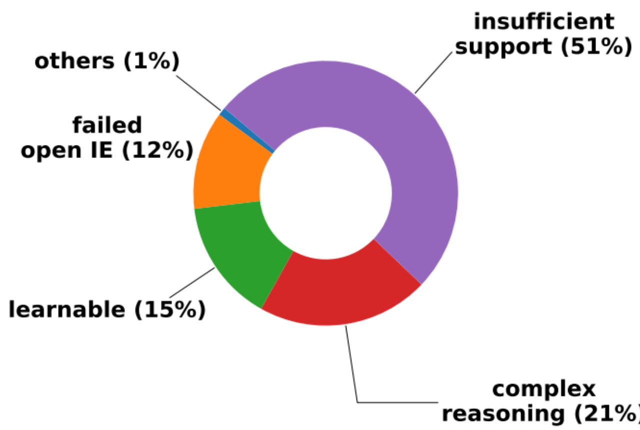
- Dataset: ARC Challenge Set. which includes 1172 questions.

- Baselines:
  - Guess-all/Random
  - IR-based algorithms: Including IR-ARC which learns on ARC corpus, and IR-Google which is on a larger corpus
  - TableILP: Formulate the reasoning as an ILP
  - Deep learning based algorithms: DecompAttn, DGEM-OpenIE, BiDAF

- Results:

Method	Test Scores
IR-ARC	20.26
IR-Google	21.58
TupleInference	23.83
DecompAttn	24.34
Guess-all / Random	25.02
DGEM-OpenIE	26.41
BiDAF	26.54
TableILP	26.97
KG^2	**31.70**

KG^2 outperforms all baselines, however is still far from human performance.



Authors believe that the inference part is performing well, however, the supporting set is not enough.

References

[1] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[2] Yuyu Zhang, Hanjun Dai, Kamil Toraman, and Le Song. Kgˆ 2: Learning to reason science exam questions with contextual knowledge graph embeddings. *arXiv preprint arXiv:1805.12393*, 2018.