Supplemental Material for " A Holistic Approach for Answering Logical Queries on Knowledge Graphs"

Yuhan Wu
East China Normal University
yuhanwu0001@163.com

Yuanyuan Xu
Zhejiang Lab,University of New South Wales
xyycn8@163.com

Xuemin Lin
Shanghai Jiao Tong University
lxue@cse.unsw.edu.au

Wenjie Zhang
University of New South Wales
wenjie.zhang@unsw.edu.au

I. The proposed HaLk

TABLE I: Notations and definitions.

Symbols	Definition
$\mathcal{G} = \{V, R, T\}$	the knowledge graph
v_i	the i-th entity/node in knowledge graph
$\boldsymbol{v}_i$	the embedding of entity/node v_i
r_i	the i-th relation/edge in knowledge graph
$\boldsymbol{e}_i$	the embedding of relation/edge r_i
$\mathcal{Q} = \{U, R, L\}$	a dependency graph
U	the node set of $\mathcal{Q}$
u_i	the i-th node in U
L	the set of Logical operations
$\boldsymbol{A}_i$	the arc embedding of U_i
$\boldsymbol{A}_{i,c}$	the center points of $\boldsymbol{A}_i$
$\boldsymbol{A}_{i,l}$	the arclengths of $\boldsymbol{A}_i$
$\boldsymbol{A}_{i,S}$	the start points of $\boldsymbol{A}_i$
$\boldsymbol{A}_{i,E}$	the end points of $\boldsymbol{A}_i$

A. Semantic Center

We illustrate it using an example in Fig. 1. Without loss of generality, let $d = 1$, when $A_{1,c} = (\rho, \alpha)$ and $A_{2,c} = (\rho, 2\pi - \beta)$ $(0 < \alpha, \beta < \pi/2)$, the common weighted average point of $A_{1,c}$ and $A_{2,c}$ may lead to wrong average center point that is closed to (ρ, π) (see Fig. 1a), yet, actually, the expected semantic center point should be around $(\rho, 0)$ (see Fig. 1b). In a word, due to the periodicity of angle, the usual weighted average may lead to inconsistent semantics.

B. Difference operation

Lemma 1: The boundary of answer region returned by HaLk is tighter than the one returned by NewLook.

Proof 1: In fact, Newlook would generate false negatives in $\boldsymbol{a}_2$ and $\boldsymbol{a}_3$ when only modelling answer area $\boldsymbol{a}_1$. If more answers are required, it would include false positives by expanding the box area to the yellow dashed rectangle, that is, covering the unexpected entities in $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$. On the other hand, HaLk is able to completely model the answer area by arcs as shown in Fig 3b since the answer area $\boldsymbol{A}_q$ of $\boldsymbol{A}_1 - \boldsymbol{A}_2 - \boldsymbol{A}_3$ is still a closure of arc. Thus, the theorem holds.

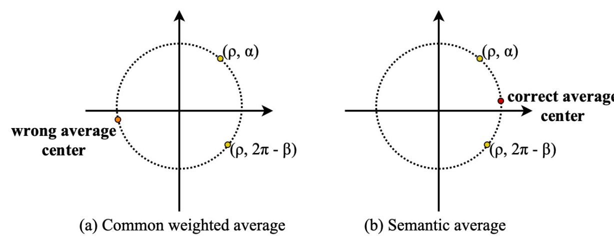

Fig. 1: Illustrations of two kinds of center average protocols.

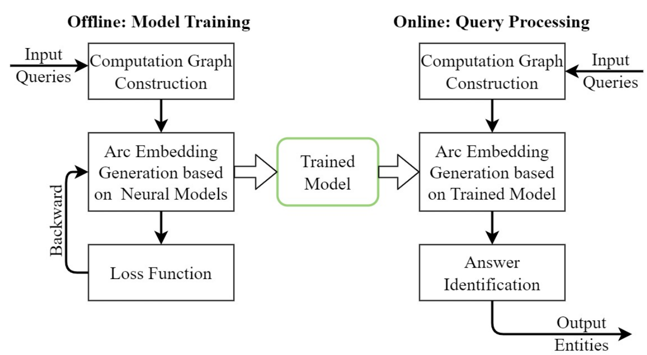

Fig. 2: Framework of our proposed HaLk.

Remarks. Here are two special situations, that is, $\boldsymbol{b}_1/\boldsymbol{A}_1$ is entirely inside $\boldsymbol{b}_i/\boldsymbol{A}_i$, or multiple boxes/arcs have no overlap. Both NewLook and HaLk can completely model answer regions via attention neural network, yet these two situations are rarely raised in real-world scenarios. For example, "*What*

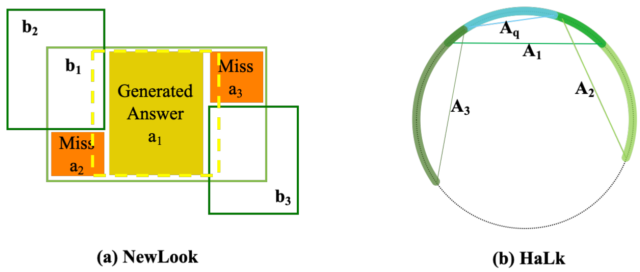

Fig. 3: Illustrations of two models for difference operator.

plants are living in the desert but not living under water?"

C. Framework of HaLk

Fig. 2 shows the overall framework of the proposed HaLk including the offline stage and online stage. During the training stage, for the input training query structures, we need to first construct the corresponding computation graph, and then traverse the computation graph starting from anchor node(s) to iteratively solve the sub-queries til get the arc embedding of the target node of the query. After that, we calculate the loss and make a gradient back propagation to update the embeddings of the entities, the embeddings of the relations and the parameters of all the logical operators' networks. The training will continue until the predetermined maximum iterative steps or model convergence is reached. As for online query stage, the trained model would generate the arc embedding for the target node of the input test logical query by executing a set of logical operations. According to the obtained arc embedding, we measure the similarity between entities and the query by the distance between them in the vector space. At last, all the entities that are inside or close to the query's arc embedding would be selected as answers to the query.

II. Experiments

A. Training and evaluation query structures.

As shown in Fig. 4, the query structures on the top line considered to verify HaLk's ability to model the newly-defined FOL operations, (i.e., projection, intersection, difference and union). And the complex query structures including 'ip', 'pi', '2u', 'up' and 'dp' would only be evaluated on validation and test stage to test the generalization performance of HaLk. The additional query structures with negation operator on the bottom line is used to evaluate model's performance on queries with negation operation together with the top line training query structures.

B. Visualization of Answering Logical Queries

To explain more clearly about how to answer a logical query on knowledge base on HaLk, we visualize the intermediate answer entities and target answer entities for a given question example *"What company was founded by a student from West Point and was also the one that Buzz Aldrin work for?"*. Since the intermediate results for the variables are also arc embeddings, we can also get the prediction results for the intermediate variables via LSH in the embedding space.

Combined with the variable nodes marked in Fig. 5, we can see from Table. II that although the prediction results of intermediate variable nodes a and b do not hit the randomly selected positive entity, the prediction results of variable node c and target node d both hit the randomly selected ground truth.

C. Computational Complexity

The computational costs of HaLk are similar to that of many embedding based methods [1]–[3] that also model logical operations as geometric shapes. During the online stage, answering a logical query is reduced to simply processing the k conjunctive queries in Eq.(1), where k might not be so large in practice. Note that all the k computations can be parallelized. Furthermore, we execute a sequence of simple logical operations based on arc embedding for each conjunctive query, each of which takes constant time. To get the final answers, we perform a range search in the low-dimensional vector space, which can also be done in constant time using search algorithms, e.g., Locality Sensitive Hashing (LSH) [4]. Overall, it is very fast to process a logical query online, which has been empirically discussed in Section *Efficiency of Query Processing*.

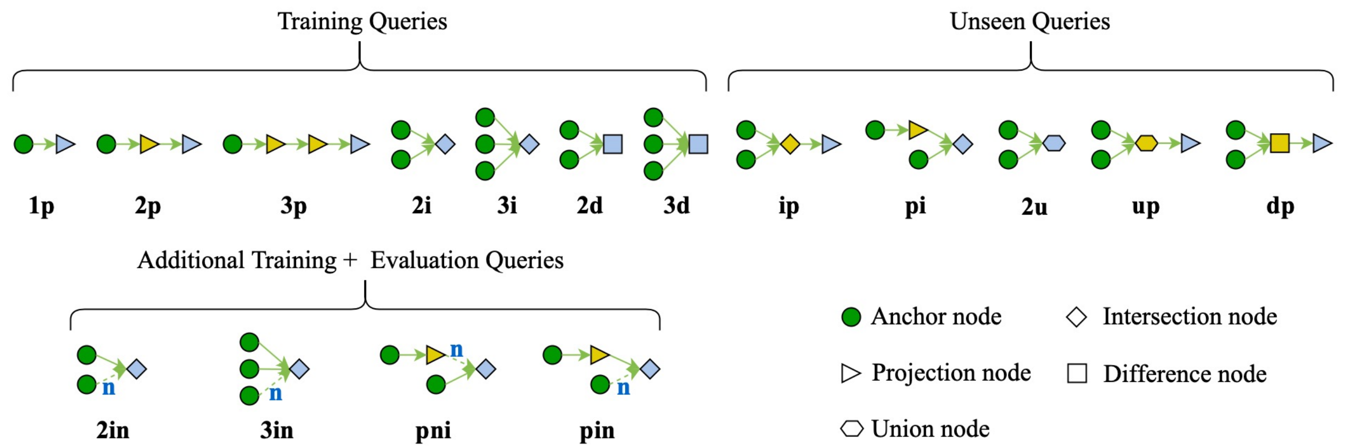

Fig. 4: Query structures with abbreviation of their computation graph used in the experiments, where 'p', 'i', 'd', 'n' and 'u' represent 'projection', 'intersection', 'difference', 'negation' and 'union'.

TABLE II: Visualization of a pi query from FB15k-237 test set.

Query	$q = d_?.\exists a : Student(West\ Point, a) \vee Found(a, d_?) \vee Company(Buzz\ Aldrin, d_?)$		
Variable	Top 6 Prediction Results		Positive Sample
a	David Randolph Scott Simone Askew Roy Moore	Roger Hilsman Jr. Frank Frederick Borman II Mark Thomas Esper	Donald Herod Peterson
b	NSA NASA Scrum	Colgate-Palmolive CIA US Department of Housing and Urban Development	US Department of the Air Force
c	US Air Force Academy Microsoft RAND Corporation	The Walt Disney Company NASA Omega SA	NASA
d	Apple NASA CBS	Pixar Animation Studio US Air Force Academy Slitherine Software	NASA

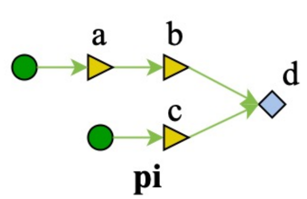

Fig. 5: Computation graph of question *"What company was founded by a student from West Point and was also the one that Buzz Aldrin work for?"*.

REFERENCES

[1] L. Liu, B. Du, H. Ji, C. Zhai, and H. Tong, "Neural-answering logical queries on knowledge graphs," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2021, pp. 1087–1097.

[2] H. Ren, W. Hu, and J. Leskovec, "Query2box: Reasoning over knowledge graphs in vector space using box embeddings," in *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[3] Z. Zhang, J. Wang, J. Chen, S. Ji, and F. Wu, "Cone: Cone embeddings for multi-hop reasoning over knowledge graphs," in *Advances in Neural Information Processing Systems*, 2021, pp. 19 172–19 183.

[4] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas*. ACM, 1998, pp. 604–613.