Appendix

I. WikiHop Benchmark

A. Dataset

Wikihop [1] is a cross-document multi-step Reading comprehension dataset. There are 43738 questions in the train dataset and 5129 questions in the validation dataset. Specifically, the original dataset doesn't provide the grounded document supervision, we process the dataset to satisfy our retriever task setting.

Based on the methodology in the dataset paper [1], they generated this multi-hop dataset by a traversal way on hyperlinked Wikipedia corpus. Each instance in the source code contains:

- a natural language query q;
- a support document set T;
- an answer a to q which can be inferred by multi-hop reasoning from multiple documents in the document set T.

For the specific retrieval task, we aim to retrieve the grounded document path P_t to the query q.

B. Ground document path Generation

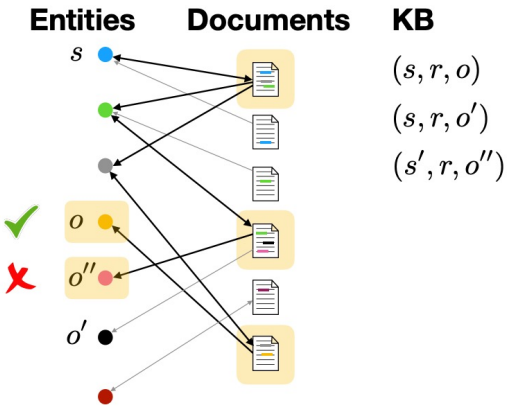

Fig. 1: A directed bipartite graph for the Wikihop query-document data generation. Vertices on the left side correspond to entities $E = \{e_1, e_2, ..., e_m\}$ in the Knowledge Base, and the right vertices refer to the Wiki documents $D = \{d_1, d_2, ..., d_n\}$. A document node d is connected to an entity e if e is mentioned in d. This picture is from [1].

For each question q, a directed bipartite graph is generated for a query as shown in Fig 1. The generation of the set of support files T starts with a seed file describing the entity $e\prime$ in q and traverses the bipartite graph using breadth-first search. They will stop the traversal at the files containing the answer entity.

In this way, we can only generate the ground hop-1 document for the multi-hop question in a semi-supervision manner, *i.e.*, the seed entity document d' is the ground hop-1 document. We conduct entity linking over the query and find the entity's wikipedia document which is contained in the provided support set T. Same to the HotpotQA training setting, we generate the 9 negatives T_n by:

1. We use BM-25 algorithm to fetch the top 10 documents T_{bm} from the open-domain corpus for each query. We generate a hard negatives set $T_b = T_{bm} - d'$.
2. For each query, we generate a source negatives set by remove the ground hop-1 document from the support document set, $T_s = T - d'$.
3 If $|T_s| == 9$, $T_n = T_s$; if $T_s < 9$, we extend T_s by T_b until $|T_s| == 9$; if $T_s > 9$, we select the top 9 documents from T_s by BM-25 score between query and document as T_n.

II. Proof of the generation problem.

Theorem 1. *The generation problem of a triple fact set C_i' without cover-redundancy is NP-hard.*

Proof. We prove the generation problem is NP-hard by a reduction from the set cover problem. An instance of set cover problem has the input of a set U of n elements, and a collection $S = \{S_1, S_2, ..., S_m\}$ of m subsets of U such that $\cup_i S_i = U$. The goal is to select as few subsets as possible from S such that their union covers U. We construct an instance of the generation as follows: given a triple fact set $C = \{t_1, t_2, ..., t_n\}$, we aim to generate a non-overlapped triple fact set C', $\forall t_i, t_j \in C'$, $s(t_i) \not\subset s(t_j)$ and $s(t_j)) \not\subset s(t_i)$. Specifically, a triple fact $t_i \in C$ can cover some triple facts $\{t_i, t_j, ..., t_k\}$ and there indeed exists a un-overlapped triple fact set of C, $\tilde{C} = \{t_1, t_2, ..., t_{n'}\}$ where $n' < n$. The un-overlapped triple fact set contains n' unique non-overlapped triple facts, $\forall t_i, t_j \in \tilde{C}$, $s(t_i) \not\subset s(t_j)$ and $s(t_j)) \not\subset s(t_i)$. We can maintain a set C of coverage sets, $C_i = \{t_i, t_j, ..., t_k\}$, we aim to generate a smallest set of C which covers $\tilde{C}_i$. Therefore, the generation problem selects the coverage set corresponding to the set that covers all the elements with as few subsets as possible, which means that each solution of the generation problem is a solution of the set cover problem. □

III. Greedy algorithm

Theorem 2. *The greedy-based algorithm has an approximation ratio of $H(n')$.*

Proof. [2] proved that the greedy algorithm to the set cover problem has an approximation ratio of $H(n')$. □

REFERENCES

[1] J. Welbl, P. Stenetorp, and S. Riedel, "Constructing datasets for multi-hop reading comprehension across documents," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 287–302, 2018.

[2] V. K. Wan and K. D. Ba, "Set cover and application to shortest superstring," https://www.cs.dartmouth.edu/~ac/Teach/CS105-Winter05/Notes/wan-ba-notes.pdf, 2005, [Online].