

Appendix: Feature Description

We provide details of the features in our learning framework. We list the description and categorization of different features, and the policies they are associated with in Table 7.

Categorization of features. To get most possible speedup, we try to use features that are cheap to obtain or computed as the part of STAR process. We can categorize our features to three categories. 1) *Static features* that are computed from the query structure and labels. As these features do not change in different states of L2P-STAR, they compute once and use for both policies. 2) *Ranking features* that are related to the score of current top-k answers, upperbound of each star query, growth of lowerbound and reduction in upperbound at the sequence of states; and 3) *Context features* that are computed based on the current state of L2P-STAR like the number of fetched partial answers from each star query and the selected star for evaluation at that state.

Case Analysis. All of the features are inherently from the nature of top- k querying or STAR approach. Intuitively, *Static features* are related to the hardness of answering a query. For example, by increasing the number of joinable nodes for each star query, finding a complete answer would be harder. Similarly, “Estimated partial answers” can demonstrate the possible number of fetches for each star query. These features have some degree of predictive power. Since they don’t change during the search process, we need dynamic features.

Ranking features like upper-bound of star queries can help the model to select a potential star query with higher score. Additionally, by considering their difference with the lowerbound, Π_{fetch} can leverage this information to predict how many potential items should be fetched. For example, if the difference between upper and lower bound is high, Π_{fetch} should fetch more to fill the gap.

A context feature like “StarkIsEnough” will provide the information that all partial answers are finished for a star query and can help Π_{select} in pruning the corresponding star queries from selection.

Features	Description	Category	Policy
# of nodes	Number of nodes in the query	static	Ψ_1, Ψ_2
# of edges	Number of edges in the query	static	Ψ_1, Ψ_2
# of stars	Number of star queries	static	Ψ_1, Ψ_2
# of star nodes	Number of nodes in each star query	static	Ψ_1, Ψ_2
AveDeg	Average degree of the candidate nodes for each star query	static	Ψ_1, Ψ_2
Estimated Partial Answers	Average number of partial answers of each star query	static	Ψ_1, Ψ_2
PriorityQueue Items	Size of the priority queue of candidate pivot nodes for each star query	static	Ψ_1, Ψ_2
JoinableNodes	Number of joinable nodes each star query has	static	Ψ_1, Ψ_2
CurrentPriorityQueue	Size of current pivot nodes in the priority queue of each star query	context	Ψ_1
CurrentUB	Current upperbound of each star query	ranking	Ψ_1
CurrentLB	Current lowerbound of answers (Score of k-th answer in top-k list)	ranking	Ψ_1
PQDiffFromParentState	The difference between the number of candidate pivot nodes in current L2P-STAR state and previous state (after fetching some items)	context	Ψ_1
UBDiffFromCurrentLB	The difference between upperbound of each star query and the lowerbound of answers	ranking	Ψ_1
UBDiffFromParentUB	The difference between current upperbound of each star query in current state of L2P-STAR and the previous state of L2P-STAR (before and after fetching)	ranking	Ψ_1
LBDiffFromRootLB	The difference between current lowerbound from the starting lowerbound. (Current lowerbound vs. lowerbound after fetching for all star queries)	ranking	Ψ_1
LBDiffFromParentLB	The difference between current lowerbound at this state of L2P-STAR and previous state of L2P-STAR. (Before and after fetching)	ranking	Ψ_1
# SelectedBefore	How many times each star query was selected before?	context	Ψ_1
CurrentDepth	How many items are fetched for each star query?	context	Ψ_1
StarkIsEnough	If there exists any remaining item or not for each star query	context	Ψ_1
remainingPA	Remaining estimated partial answers for each star query	context	Ψ_1
previousSelectedSQ	The index of previous selected star query for fetching	context	Ψ_1
CurrentPriorityQueue	Current size of priority queue for this star query	context	Ψ_2
CurrentUpperbound	Current upperbound for this star query (The best score that we may get by fetching for this star query)	ranking	Ψ_2
CurrentLowerbound	Current lowerbound of the answers (Score of k-th answer in the current top-k list)	ranking	Ψ_2
UBDiffFromCurrentLB	The difference between upperbound of the star query and the lowerbound of answers	ranking	Ψ_2
UBDiffFromRootUB	The difference between upperbound of the star query and the upperbound of the same star query after fetching the first items	ranking	Ψ_2
LBDiffFromRootLB	The difference between lowerbound of the star query at this state and the lowerbound of the answers after fetching the first items	ranking	Ψ_2
CurrentDepth	How many items are fetched from this star query before?	context	Ψ_2
StarkIsEnough	If there exists any remaining item for each star query	context	Ψ_2
RemainingPA	Estimation of remaining partial answers for this star query	context	Ψ_2
SearchLevel	Number of states in the path so far	context	Ψ_2
IsPreviouslySelected	Is this star query also selected in the previous state?	context	Ψ_2

Table 7: Complete details of all features including their definition, categorization, and their association with Ψ_1 and Ψ_2 .