# Indoor Top-$k$ Keyword-aware Routing Query


Zijin Feng[†] Tiantian Liu[‡] Huan Li[‡] Hua Lu[‡] Lidan Shou[§] Jianliang Xu[†]

[†]Department of Computer Science, Hong Kong Baptist University, Hong Kong

[‡]Department of Computer Science, Aalborg University, Denmark

[§]Department of Computer Science, Zhejiang University, China


## 1. Introduction

**Applications of Indoor Keywords Route planning.**

*Daily life*: people go through a large and/or unfamiliar indoor environment.

*Industry*: indoor robots can make use of indoor routing with keywords to accomplish operational tasks.

**An indoor top-k keyword-aware routing query (IKRQ) is non-trivial.**

Define keyword relevance for routes w.r.t. query keywords.

Integrate keyword relevance and spatial distance for ranking routes.

Search for routes in an indoor venue with a large number of partitions that form complex topology, which may result in a large search space for routing.

**To resolve IKRQs, we develop a set of techniques.**

The concept of prime routes diversifies top-k routes in the query result.

Bi-directional mapping structures organize two level indoor keywords.

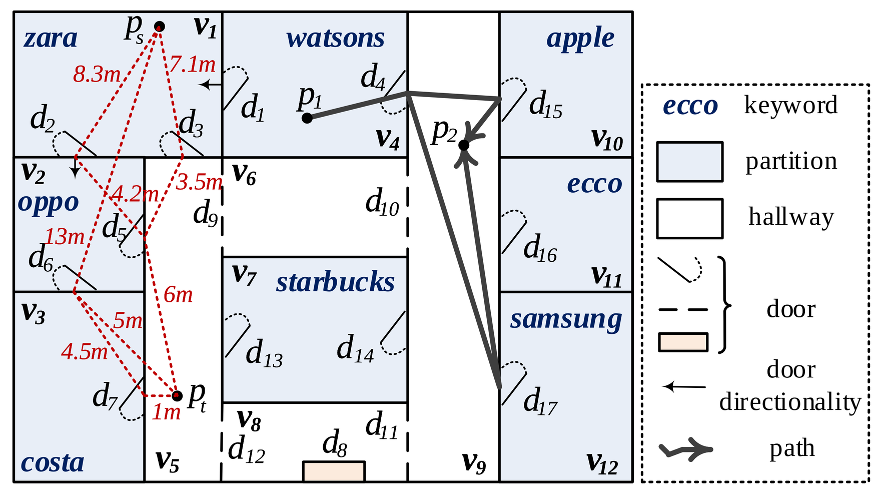
Several other pruning rules are derived based on the distance constraint and the bound of top-k result.

Two search algorithms (ToE and KoE) are designed for routing.

## 2. Problem Formulation

**Homogeneous Routes.** Two routes $R_i$ and $R_j$ are ***homogeneous routes*** if $R_i.head = R_j.head$, $R_i.tail = R_j.tail$, and $KP(R_i) = KP(R_j)$.

**Prime Route.** Suppose $HR$ is a complete set of homogeneous routes for a routing query, we say a route $R_i \in HR$ is ***prime*** against $R_j \in HR$ if $\delta(R_i) < \delta(R_j)$. $R_i$ is a ***prime route*** if $R_i$ is prime against all other routes in $HR$.

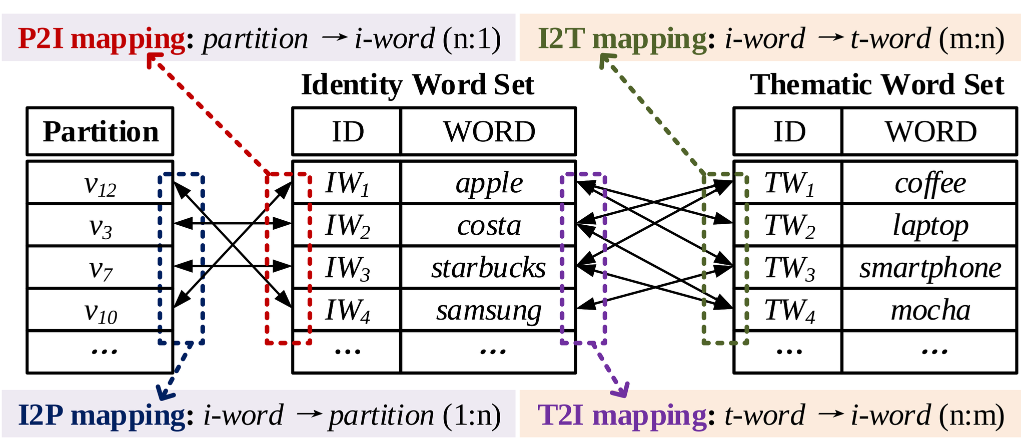



**Indoor Top-$k$ Keyword-aware Routing Query.** Given a start point $p_s$, a terminal point $p_t$, a distance constraint $\Delta$, and a query keyword list $QW$, an ***indoor top-$k$ keyword-aware routing query*** IKRQ($p_s$, $p_t$, $\Delta$, $QW$, $k$) returns $k$ regular and prime routes from $p_s$ to $p_t$ in a $k$-set $\Theta$ such that $\forall R \in \Theta$, $\delta(R) \le \Delta$ and $\Psi(R, \Delta, QW) \ge \Psi(R', \Delta, QW)$ for any route $R' \notin \Theta$ from $p_s$ to $p_t$ with $\delta(R') \le \Delta$.

## 3. Summary of Our Approach

### 3.1 Relevance Score.

***Identity Word and Thematic word.*** An **identity word** (i-word) identifies the specific name of a partition, while a **thematic word** (t-word) refers to a tag relevant to that partition.




***Keyword Relevance.***

$$\rho_{QW}(R) = \begin{cases} 0, & \text{if } N_{QW}(R) = 0; \\ N_{QW}(R) + \dfrac{\sum_{w_Q \in QW} \left( \max_{w'_i \in M(w_Q, R)} s(w'_i) \right)}{N_{QW}(R)}, & \text{otherwise.} \end{cases}$$

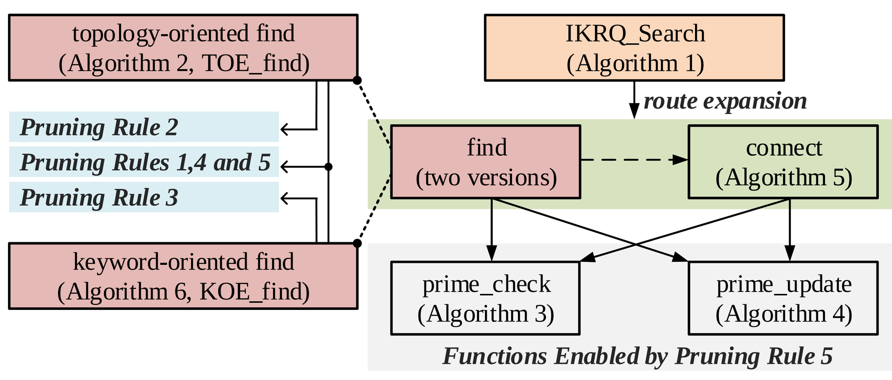
***Ranking Score.***

$$\psi(R, \Delta, QW) = \alpha \cdot \frac{\rho(R)}{|QW| + 1} + (1 - \alpha) \cdot \left(\frac{\Delta - \delta(R)}{\Delta}\right)$$

### 3.2 Overall Search Framework and Pruning Rules.

**Pruning Rule 1:** A partial route $R^\star = (p_s, d_i, \ldots, d_n)$ in the searching can be pruned if $\delta(R^\star) + |d_n, p_t|_L > \Delta$.

**Pruning Rule 2:** A door $d_n$ can be pruned out of the search if $|p_s, d_n|_L + |d_n, p_t|_L > \Delta$.




**Pruning Rule 3:** An indoor partition $v_i$ can be pruned out of the search if its lower bound distance $\delta(p_s, v_i, p_t)$ =

$$\min_{d_i \in P2D_{\sqsupset}(v_i), d_j \in P2D_{\sqsubset}(v_i)} (|p_s, d_i|_L + \delta_{d2d}(d_i, d_j) + |d_j, p_t|_L) > \Delta.$$

**Pruning Rule 4:** Given the current $k$-th highest ranking score $\psi_k$ among the seen complete routes, a partial route $R^\star = (p_s, d_i, \ldots, d_n)$ can be pruned if its upper bound ranking score $\psi_U(R^\star) = \alpha \cdot 1 + (1 - \alpha)(1 - (\delta(R^\star) + |d_n, p_t|_L)/\Delta) \le \psi_k$.

**Pruning Rule 5:** A partial route $R^\star = (p_s, d_i, \ldots, d_n)$ in the search can be pruned if the search has already obtained a route $R^{\star'}$ from $p_s$ to $d_n$ that is prime against $R^\star$.

### 3.3 ToE and KoE.

**ToE:** ToE always expands from the current door to the next enterable door within one hop.

**KoE:** KoE focus on the query words that have not been covered by the current stamp, and directly expand to one of the key partitions that can cover some of those uncovered query words.

## 4. Experimental Results

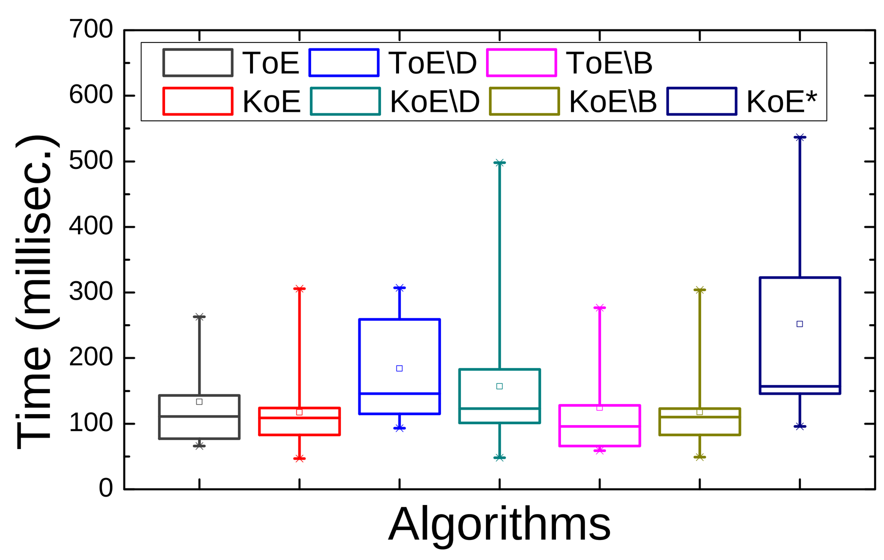
### 4.1 Results on Synthetic Data.

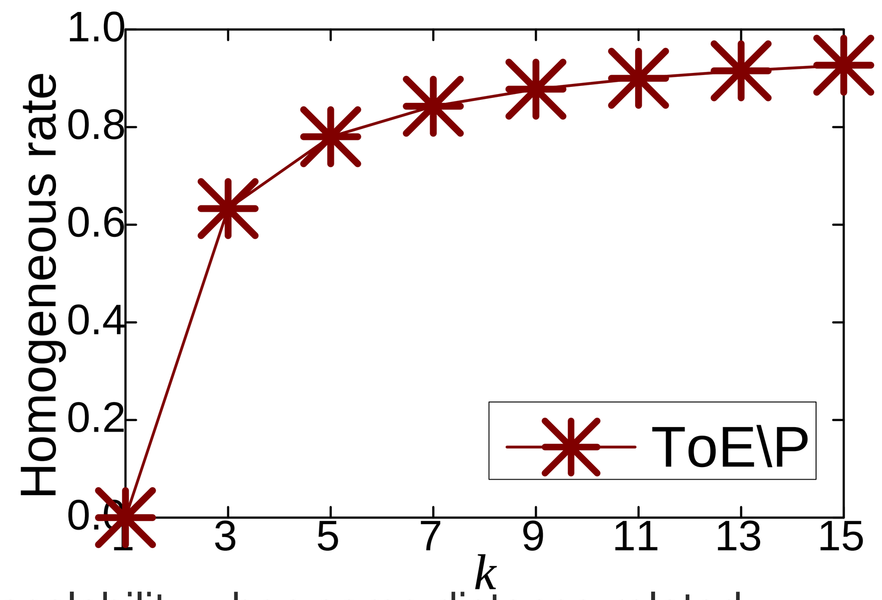
**Indoor Space Settings.** Based on a real-world floorplan, we generate a multi-floor indoor space where each floor takes 1368m × 1368m with 96 rooms, 4 hallways, and 4 staircases. As a result, we obtain 141 partitions and 220 doors on each floor. We duplicate the floorplan 3, 5, 7, or 9 times to simulate different indoor spaces.

**Parameter Settings.**

| Parameters | Settings |
|---|---|
| $k$ | 1, . . . , **7**, . . . , 11 |
| $\lvert QW\rvert$ | 1, 2, 3, **4**, 5 |
| $\beta$ (% of i-words in $QW$) | 20%, 40%, **60%**, 80%, 100% |
| $\delta_{s2t}$ (meter) | 1100, 1300, **1500**, . . . , 2100 |
| $\eta$ | 1.4, **1.6**, 1.8, 2.0 |
| $\alpha$ | 0.1, 0.3, **0.5**, 0.7, 0.9 |
| $\tau$ | 0.05, **0.1**, 0.2, 0.4 |

**Indoor Keywords Settings.** We use Scrapy to crawl the online shop information from five shopping malls in Hong Kong, obtaining 2074 documents for 1225 shop brands. All the 1225 brand names are used as i-words. For each such i-word, we use up to 60 extracted keywords with the highest TF-IDF values as its t-words.In total, we have 9195 t-words and each i-word corresponds to 16.6 t-words on average. We randomly assign an i-word and all its t-words to each room.

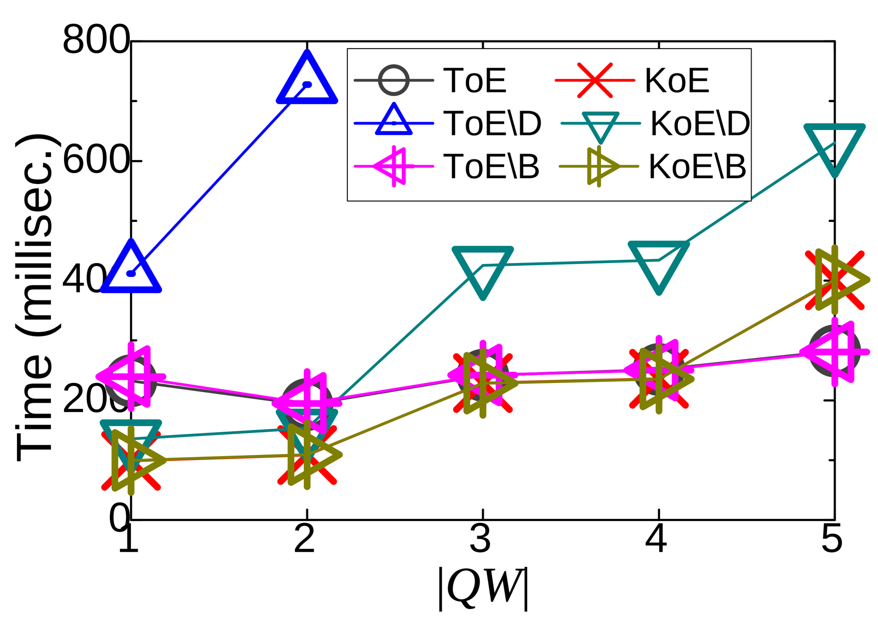


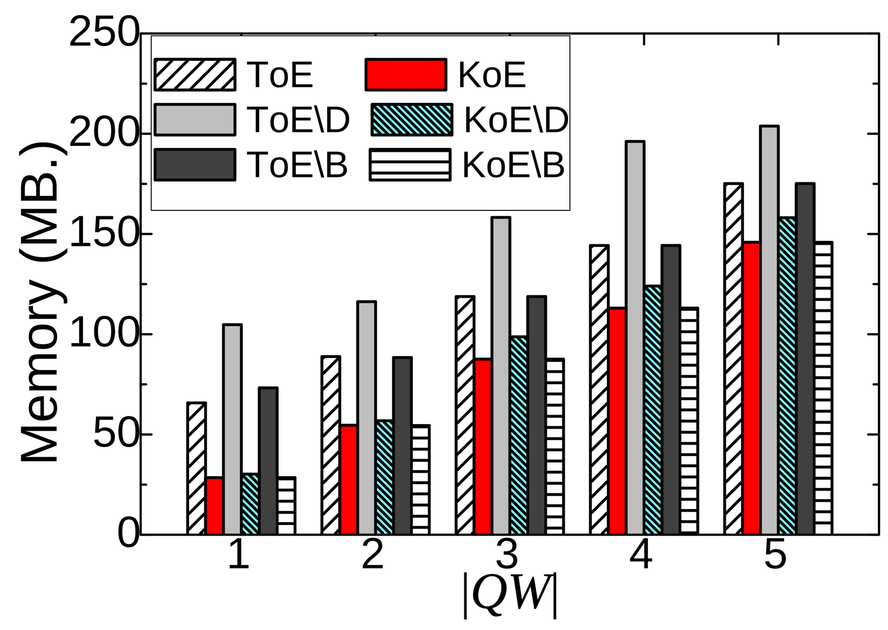




**Result Summary:** In general, KoE has better scalability when some distance-related parameters (e.g., $\eta$ and $\delta_{s2t}$) are enlarged. Conversely, ToE is more efficient when there are more query words. In addition, KoE always has a lower memory cost.

### 4.2 Results on Real Data.

**Settings.** We collect a dataset with real indoor topology and keyword distributions from a seven-floor, 2700m × 2000m shopping mall in Hangzhou, China. We extract the keywords from the store descriptions on the mall's website and obtain 5036 t-words for 533 i-words (stores). There are 103 stores with no t-words but only one i-word. An i-word corresponds to 31 t-words maximum and 9.4 ones on average. **Results.**





❇ Accepted by @ ICDE'20 | ✉ liutt@cs.aau.dk