Indoor Top-$k$ Keyword-aware Routing Query

Zijin Feng $^1$ & Tiantian Liu $^2$ & Huan Li $^2$ & Hua Lu $^2$ & Lidan Shou $^3$ & Jianliang Xu $^1$

$^1$Department of Computer Science, Hong Kong Baptist University
$^2$Department of Computer Science, Aalborg University
$^3$Department of Computer Science, Zhejiang University

April 2020
1 Motivation
2 High-Level Overview
3 Indoor Topology & Route
4 Keyword Organization & Relevance
5 Search Algorithms for IKRQ
6 Experimental Studies
This paper:
- Formulates indoor top-$k$ keyword-aware routing query (IKRQ)
- Devises mapping structures to organize indoor keywords and compute route keyword relevance
- Derives pruning rules to reduce search space in routing
- Conducts extensive experiments on synthetic and real data sets to evaluate our proposals
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 $\text{IKRQ}(p_s, p_t, \Delta, QW, k)$ returns $k$ regular and prime routes from $p_s$ to $p_t$ in a $k$-set $\Theta$ such that $\forall R \in \Theta$, $\delta(R) \leq \Delta$ and $\Psi(R, \Delta, QW) \geq \Psi(R', \Delta, QW)$ for any route $R' \notin \Theta$ from $p_s$ to $p_t$ with $\delta(R') \leq \Delta$. 

Motivation
Figure: Architecture of the IKRQ Search Algorithms
Homogeneous Routes. Two routes $R_i$ and $R_j$ are **homogeneous routes** if $R_i.\text{head} = R_j.\text{head}$, $R_i.\text{tail} = R_j.\text{tail}$, 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 $HR$.

**Principles of indoor route search**

- **Principle of Regularity.** Disqualifies a route that contains one or more doors between two identical doors (e.g. d13-d14-d14-d13)
- **Principle of Diversity.** Avoid homogeneous routes in our indoor routing

| TABLE II: Examples of Routes from $p_s$ to $p_t$ |
|-----------------|---------------------------------|
| $R_1$           | $(p_s \xrightarrow{v_1} d_2 \xrightarrow{v_3} d_6 \xrightarrow{v_3} d_7 \xrightarrow{v_3} p_t)$ |
| $R_2$           | $(p_s \xrightarrow{v_1} d_2 \xrightarrow{v_3} d_5 \xrightarrow{v_3} d_7 \xrightarrow{v_3} p_t)$ |
| $R_3$           | $(p_s \xrightarrow{v_1} d_2 \xrightarrow{v_3} d_5 \xrightarrow{v_3} d_9 \xrightarrow{v_3} d_7 \xrightarrow{v_3} d_7 \xrightarrow{v_3} p_t)$ |
| $R_4$           | $(p_s \xrightarrow{v_1} d_3 \xrightarrow{v_3} d_5 \xrightarrow{v_3} d_7 \xrightarrow{v_3} d_7 \xrightarrow{v_3} p_t)$ |
Identity word (i-word). Identifies the specific name of a partition (e.g. Starbucks)

Thematic word (t-word). Refers to a tag relevant to that partition (e.g. coffee, mocha)

**Figure: Indoor Space Keyword Mappings**

For T2I mapping, we have **direct matching i-words** and **indirect matching i-words**
### Candidate I-word Set $\kappa(w_Q)$

Set of entries each of which is in form of $(w_i, s)$, a pair of a matching i-word $w_i$ and the similarity score $s$ between $w_Q$ and $w_i$, $s > \tau$.

$\kappa(w_Q)$ has two cases:

- If $w_Q$ is an i-word, $\kappa(w_Q) = \{(w_Q, 1)\}$
- If $w_Q$ is a t-word, $\kappa(w_Q)$ consists of
  - All direct matching i-word, i.e., $(w'_i, 1)$, for all $w'_i \in T2I(w_Q)$
  - All indirect matching i-word, i.e., $(w''_i, s(w''_i))$, where
    \[
    s(w''_i) = \frac{|I2T(w''_i) \cap \bigcup_{w_i \in T2I(w_Q)} I2T(w_i)|}{|I2T(w''_i) \cup \bigcup_{w_i \in T2I(w_Q)} I2T(w_i)|} > \tau
    \]

<table>
<thead>
<tr>
<th>partition</th>
<th>i-word</th>
<th>t-words</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_3$</td>
<td>costa</td>
<td>{coffee, drinks, macha}</td>
</tr>
<tr>
<td>$v_{10}$</td>
<td>apple</td>
<td>{phone, mac, laptop, watch}</td>
</tr>
<tr>
<td>$v_7$</td>
<td>starbucks</td>
<td>{coffee, macha, latte, drinks}</td>
</tr>
<tr>
<td>$v_{12}$</td>
<td>samsung</td>
<td>{phone, laptop, earphone}</td>
</tr>
</tbody>
</table>

### Example

$I2T(costa) = \{coffee, drinks, macha\}$ and $\bigcup_{w_i \in T2I(latte)} I2T(w_i) = \{coffee, drinks, macha, latte\}$
Keyword Relevance.

\[
\rho_{QW}(R) = \begin{cases} 
0, & \text{if } N_{QW}(R) = 0; \\
N_{QW}(R) + \frac{\sum_{w_Q \in QW} \left( \max_{w_i' \in M(w_Q, R)} s(w_i') \right)}{N_{QW}(R)}, & \text{otherwise.}
\end{cases}
\]

Ranking Score.

\[
\psi(R, \Delta, QW) = \alpha \cdot \frac{\rho(R)}{|QW| + 1} + (1 - \alpha) \cdot \left( \frac{\Delta - \delta(R)}{\Delta} \right)
\]
Search Algorithms for IKRQ

Pruning rules

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

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

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

$$\min_{d_i \in P2D_{\sqcup}(v_i), d_j \in P2D_{\sqcap}(v_i)} ((|p_s, d_i|_L + \delta d_2 d(d_i, d_j) + |d_j, p_t|_L) > \Delta.$$ 

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

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

Topology-oriented Expansion (ToE)

Idea: To reach all accessible doors from the current door based on indoor topology, i.e., always expands from the current door to the next enterable door within one hop

Keyword-oriented Expansion (KoE)

Idea: 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

**Algorithm 1** IKRQ_Search ($p_s$, $p_t$, $A$, $QW$, $k$)

1: initialize priority queue $Q$
2: set of all candidate i-words $W_c \leftarrow \bigcup_{w \in QW} \kappa(w) \cdot W_i$
3: $P \leftarrow \left( \bigcup_{w \in QW} I2P(\kappa(w) \cdot W_i) \right) \setminus v(p_s) \cup v(p_t)$
4: door sets $D_h \leftarrow \emptyset$, $D_f \leftarrow \emptyset$
5: $kbound \leftarrow 0$
6: initialize hashtable $H_{\text{prime}}$
7: $R_0 \leftarrow \{p_s\}$
8: $S_0 \leftarrow v(p_s), R_0, 0, \rho(R_0), \psi(R_0)$
9: $Q.push(S_0)$
10: while $Q$ is not empty do
11: $S_i \leftarrow Q.pop()$
12: $ES \leftarrow \text{find}(S_i)$
13: for each $S_j \in ES$ do
14: $\text{connect}(S_j)$
15: return current top-$k$ results

Functions Enabled by Pruning Rule 5

- I2P
- I2P(\kappa(w) \cdot W_i)
- ES
- connect
- prime_check
- prime_update

**Functional Diagram**

- **topology-oriented find** (Algorithm 2, TOE_find)
- **keyword-oriented find** (Algorithm 6, KOE_find)
- **route expansion**
- **IKRQ_Search** (Algorithm 1)
- **Pruning Rule 2**
- **Pruning Rules 1, 4, and 5**
- **Pruning Rule 3**
Experimental Studies

Table: Dataset Information (Indoor Keywords)

<table>
<thead>
<tr>
<th></th>
<th>Synthetic Data</th>
<th>Real Data</th>
</tr>
</thead>
<tbody>
<tr>
<td># of i-word</td>
<td>1120</td>
<td>533</td>
</tr>
<tr>
<td># of t-word</td>
<td>9195</td>
<td>5036</td>
</tr>
</tbody>
</table>

Table: Notations of Comparable Methods

<table>
<thead>
<tr>
<th>Modification</th>
<th>ToE family</th>
<th>KoE family</th>
</tr>
</thead>
<tbody>
<tr>
<td>–</td>
<td>ToE</td>
<td>KoE</td>
</tr>
<tr>
<td>no distance-based Pruning Rules 1 3</td>
<td>ToE\D</td>
<td>KoE\D</td>
</tr>
<tr>
<td>no k-bound-based Pruning Rule 4</td>
<td>ToE\B</td>
<td>KoE\B</td>
</tr>
<tr>
<td>no prime-based Pruning Rule 5</td>
<td>ToE\P</td>
<td>–</td>
</tr>
<tr>
<td>with precomputed shortest routes</td>
<td>–</td>
<td>KoE*</td>
</tr>
</tbody>
</table>
Experimental Studies

Fig. 6: Time vs. \(|QW|\)

Fig. 7: Memory vs. \(|QW|\)

Fig. 8: Time vs. \(\eta\)

Fig. 9: Memory vs. \(\eta\)

Fig. 10: Time vs. \(\beta\)

Fig. 11: Time vs. floor

Fig. 12: Time vs. \(\delta_{K_{2t}}\)

Fig. 13: Time of KoE*

Fig. 14: Memory of KoE*

Fig. 15: Time of ToE/P

Fig. 16: Homogeneous rate

Fig. 17: Time vs. \(|QW|\)

Fig. 18: Memory vs. \(|QW|\)

Fig. 19: Time vs. \(\eta\)

Fig. 20: Homogeneous rate
Thank you!

Thank you!