Pairwise Location-Aware Publish/Subscribe for Geo-Textual Data Streams

YING ZHONG\(^1\), SHUNZHI ZHU\(^1\), YAN WANG\(^1\), JIANMIN LI\(^1\), XINXIN ZHANG\(^1\), AND JEDI S. SHANG\(^2\)

\(^1\)School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
\(^2\)Thinvent Digital Technology Company Ltd., Shenzhen 330096, China

Corresponding author: Shunzhi Zhu (zhusz99@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672442 and Grant 61806173, in part by the Joint Funds of 5th Round of Health and Education Research Program of Fujian Province under Grant 2019-WJ-41, in part by the Science and Technology Planning Project of Fujian Province under Grant 2020 and Grant 2017J01783, and in part by the Science and Technology Planning Project of Quanzhou under Grant 2020C074.

ABSTRACT The continued proliferation of location-based social media and ridesharing services brings up the omnipresence of geo-textual data, which is often characterized by high arrival rates, diversified and duplicated content. A host of existing studies focus on efficient processing of continuous queries over geo-textual data streams by developing effective location-aware publish/subscribe systems. However, these systems often suffer from two limitations: duplicated feedings and low-quality feeding information. To address the limitations, we propose to develop a novel location-aware publish/subscribe system that feeds subscribers with geo-textual object pairs rather than individual objects. We believe that delivering object pairs to subscribers is capable of improving the information feeding quality and subscriber satisfaction. To address the problem, we propose a two-phase subscription matching scheme. The first step is online geo-textual object join. We apply a time-based sliding window that filters out out-dated geo-textual objects over the data stream. Based on the sliding window, we develop an efficient hierarchical geo-textual object online join algorithm that merges duplicated geo-textual objects. The second step is object pair matching. For each object pair, we take it as input and run a dedicated geo-textual object pair matching algorithm to find a subset of subscriptions that matches the input object pair. Our empirical study shows that our proposal is able to achieving high efficiency and effectiveness in comparison with baseline.

INDEX TERMS Publish, subscribe, geo-textual, stream.

I. INTRODUCTION

Over the recent decades, geo-textual objects are being generated at an increasing speed. Such data consists of three components: geographical location (coordinate with longitude and latitude or semantic location), text description, and timestamp indicating the arrival time of an object. There are more than 300 million monthly active users from Twitter who post 500 million tweets per day [8], [47]. Other examples include check-data data from Foursquare, business POIs from Yelp, and hotels from Booking.com. Geo-textual objects often have the following characteristics:

- **High arrival rates.** Geo-textual data can be modeled as a stream of items. Millions of web and mobile users from different social media platforms are posting their messages. As a result, the arrival rate of geo-textual data can be very high.
- **Diversified content.** Semantic meanings of each geo-textual data cover a wide range of topics, including but not limited to, general public concerns, bursty events, emotional outlet, recommendations, and comments of a particular event or person.
- **Duplicated objects.** Geo-textual data stream may contain objects with duplicated content and same geographical location.

It is of great interest to enable web and mobile users to be feeded with up-to-date geo-textual objects satisfying their personalized requirements. As such, a host of existing studies aim at developing effective location-aware publish/subscribe systems that are capable of feeding a large number of subscribers with their targeted geo-textual objects over a geo-textual data stream. Web and mobile users may register a
location-aware subscription, which consists of a location and a keyword predicate. Specifically, a location predicate can be a geographical point (e.g., $1°17′N$, $51°34′E$), a semantic location (e.g., Marina Bay Sands Garden Park, Singapore), or a region (e.g., no more than 10km from Marina Bay Sands Garden Park). A keyword predicate can be keywords connected by AND, OR, and NOT semantics, or a set of keywords evaluated by some particular text similarity metrics.

Existing location-aware publish/subscribe systems, as illustrated by Figure 1, take a stream of geo-textual objects and user registered subscriptions as input. Subscriptions are indexed by a purposeful subscription index. When a new geo-textual object arrives, the system initiates a subscription matching process, which finds all subscriptions that match the new object. To determine whether the new object matches a subscription, the systems consider spatial proximity between the object location and the subscription location, the text similarity between the object text description and the subscription keywords, and the freshness (arrival time) of the geo-textual object. Finally, the object is delivered to corresponding matching subscriptions.

![Figure 1. Framework of a location-aware publish/subscribe system.](image)

However, these location-aware publish/subscribe systems have the following limitations:

1. **Duplicated feedings.** Traditional location-aware publish/subscribe systems determine whether an object matches a subscription solely based on the relationship (e.g., spatial proximity, text similarity) between the object and the subscription. Such matching scheme may result in duplicated object delivery. For example, when a subscriber registers a subscription at Lisbon, Portugal, at August 23rd, 2020, with keywords “Bayern, PSG”, the subscription may be continuously fed with scoring messages of the Champions League final 2020 between Bayern Munich and Paris-Saint Germain. Among these messages, many of them are considered to be duplicated.

2. **Low-quality information.** One of the major nature of geo-textual data streams is that their textual content consists of noise and poorly-organized information, which may inevitably do harm to the user experience. To be more specific, the information of geo-textual objects may not be related to a real-life events due to their low-quality content, including but not limited to, typos, very short and ungrammatical sentences, and informal words. As such, subscribers are likely to have trouble in understanding the content of objects.

3. **Failure of capturing inter-object relations.** Existing location-aware publish/subscribe framework is single-item oriented, which means that the matching scheme is only based on the similarity between a subscription and a individual object. As such, it is incapable of capturing the relationship between different objects when determining whether an object matches a subscription.

To address the above three limitations of existing location-aware publish/subscribe systems, we take the first step to develop a pairwise location-aware publish/subscribe system over a geo-textual data stream. In particular, the system supports location-based subscriptions to continuously receive up-to-date relevant geo-textual object pairs from a stream of geo-textual objects. Our matching mechanism takes the following aspects into account: (1) Text similarity between each new object pair and subscription; (2) Aggregate spatial proximity between the locations of new object pair and subscription; (3) Aggregated freshness of the new object pair. Our pairwise location-aware publish/subscribe framework has the following advantages.

1. **Diversified subscription matching result.** Different from existing single-item oriented location-aware publish/subscribe framework that may feed subscribers with duplicated geo-textual objects, our pairwise publish/subscribe framework is capable of eliminating duplicated subscription matching result. As a consequence, subscribers can be feded with diversified information rather than be overwhelmed by a huge amount of duplicated information.

2. **Informative delivery.** Because that our pairwise publish/subscribe framework employs object-pair oriented delivery, the negative effect incurred by geo-textual objects with low-quality content can be alleviated by iterative object join operations. In addition, subscribers are able to get the inter-object relations through object-pair oriented delivery.

Developing a high-performance pairwise location-aware publish/subscribe system has the following challenges. First, the geo-textual object pairs should be generated in a real-time fashion. We need to develop an efficient online geo-textual object join algorithm that is capable of generating high-quality geo-textual object pairs within a very short period of time. Second, the number of subscriptions can be very large. Our pairwise location-aware publish/subscribe system is required to support millions or even tens of millions of subscriptions over a stream of geo-textual objects. Third, we are witnessing the high arrival rate of geo-textual data streams. As such, it is important for our subscription matching algorithm to handle geo-textual data streams with high throughput.

To address the aforementioned three challenges, we propose a two-phase subscription matching scheme. Specifically, the first step is online geo-textual object join. We apply a time-based sliding window that filters out out-dated geo-textual objects over the data stream. Based on the sliding window, we develop an efficient hierarchical geo-textual object online join algorithm that merges duplicated geo-textual objects. The second step is object pair matching. For each object pair, we take it as input and run a dedicated geo-textual object pair matching algorithm to find a subset of
subscriptions that matches the input object pair. In summary, our contribution is presented as follow.

- We define the new problem of pairwise location-aware publish/subscribe, which continuously feed each subscription with up-to-date geo-textual object pairs based on the personalized location and keyword requirements.
- We develop a high-performance pairwise location-aware publish/subscribe system, which consists of a sliding window based incremental online geo-textual object join algorithm and an efficient object pair matching algorithm.
- We conduct extensive experimental study on two real-life geo-textual datasets. Our experiment results show that our proposed hierarchical online object join algorithm and object pair matching algorithm are capable of achieving high efficiency and high effectiveness.

The rest of the paper is organized as follows. Section II defines our problem. Section III details our solutions to processing our subscriptions. Section V presents our experimental studies. Section VI reviews the related proposals. Finally, Section VII concludes the paper.

II. PROBLEM STATEMENT

In this section, We define the geo-textual object, the problem of hierarchical online geo-textual object join, location-aware subscription, and the problem of geo-textual object pair matching. We define geo-textual object by following an existing study [5] widely adopted by follow-up work.

Definition 1 (Geo-Textual Object): A geo-textual object is defined as \( o = \langle \psi, \rho, t_c \rangle \), where \( \psi \) is text description, \( \rho \) denotes a geographical location, and \( t_c \) represents the object timestamp.

Here, we consider the input of geo-textual data streams, where the geo-textual objects are arriving in a streaming manner. Geo-textual data streams are ubiquitous in web applications, such as geo-tagged tweets, check-ins from Tripadvisor, and comments from Booking.com.

Let \( W \) be a sliding window, \( W(D) \) be a collection of geo-textual objects within window \( W \). We proceed to present the definitions of Online Geo-Textual Object Join (OGT-Join).

Definition 2 (OGT-Join): Given a collection of geo-textual objects \( W(D) \) and a similarity threshold \( \theta \), the OGT-Join problem finds all object pairs \( \langle o_i, o_j \rangle \) in \( W(D) \) such that the similarity between \( o_i \) and \( o_j \), denoted by \( \text{Sim}(o_i, o_j) \), is no less than \( \theta \).

The similarity between two geo-textual objects is computed based on the spatial proximity, text similarity, and temporal proximity. Given objects \( o_i \) and \( o_j \), the spatial proximity, text similarity, and temporal proximity between \( o_i \) and \( o_j \) are calculated by Equation 1, Equation 2, and Equation 3, respectively.

\[
\text{SP}(o_i, o_j) = 1 - \frac{\text{dist}(o_i, \rho, o_j, \rho)}{\text{dist}_{\text{max}}}, \quad (1)
\]

where \( \text{dist}(o_i, \rho, o_j, \rho) \) denotes the Euclidean distance between \( o_i, \rho \) and \( o_j, \rho \), and \( \text{dist}_{\text{max}} \) denotes the maximum possible distance between any two locations in the underlying space.

\[
\text{TR}(o_i, o_j) = \frac{|o_i, \psi \cap o_j, \psi|}{|o_i, \psi \cup o_j, \psi|}, \quad (2)
\]

where \( |o_i, \psi \cap o_j, \psi| \) denotes the cardinality of the intersection between \( o_i, \psi \) and \( o_j, \psi \), and \( |o_i, \psi \cup o_j, \psi| \) denotes the cardinality of the union between \( o_i, \psi \) and \( o_j, \psi \).

\[
\text{TP}(o_i, o_j) = 1 - \frac{|o_i, t_c - o_j, t_c|}{\text{W}_{\text{max}} - \text{W}_{\text{min}}}, \quad (3)
\]

where \( \text{W}_{\text{max}} \) and \( \text{W}_{\text{min}} \) represent the latest timestamp and the earliest timestamp of sliding window \( W \), respectively. The aggregated similarity between \( o_i \) and \( o_j \), denoted by \( \text{AS}(o_i, o_j) \), is computed by Equation 4.

\[
\text{AS}(o_i, o_j) = \alpha_s \times \text{SP}(o_i, o_j) + \beta_s \times \text{TR}(o_i, o_j) + (1 - \alpha_s - \beta_s) \times \text{TP}(o_i, o_j). \quad (4)
\]

where \( \alpha_s \) and \( \beta_s \) are preference parameters that balance the weights of spatial proximity and text similarity, respectively. Note that \( \alpha_s + \beta_s \leq 1 \).

After having the join result, we take each object pair as the input and run the object matching algorithm, which addresses the problem of Geo-Textual Object Pair Matching (GTOPM).

Definition 3 [Region-Based Location-Aware Subscription (RLS)]: A region-based location-aware subscription \( s \) is defined as a triple \( s = (P, r) \), where \( P \) denotes a keyword predicate and \( r \) denotes a circular subscription region. An object pair \( \langle o_i, o_j \rangle \) matches \( s \) iff. \( o_i, \psi \cap o_j, \psi \) satisfies \( s, P \) and both \( o_i, \rho \) and \( o_j, \rho \) are located within \( s, r \).

Definition 4 [Score-Based Location-Aware Subscription (SLS)]: A score-based location-aware subscription \( s \) is defined as a triple \( s = (\psi, \rho, \phi) \), where \( \psi \) denotes a set of subscription keywords, \( \rho \) is the subscription location (geographical point), and \( \phi \) denotes relevance threshold. An object pair \( \langle o_i, o_j \rangle \) matches \( s \) iff. the relevancy score between \( o_i, o_j \) and \( s \), denoted by \( \text{RS}(o_i, o_j, s) \), is no less than \( \phi \). Here, \( \text{RS}(o_i, o_j, s) \) is computed by Equation 5.

\[
\text{RS}(o_i, o_j, s) = \alpha_r \times \text{TR}(\langle o_i, o_j \rangle, s) + (1 - \alpha_r) \times \text{SR}(\langle o_i, o_j \rangle, s). \quad (5)
\]

Note that \( \text{TR}(\langle o_i, o_j \rangle, s) \) and \( \text{SR}(\langle o_i, o_j \rangle, s) \) denote the text relevance and spatial relevance, respectively, between object pair \( \langle o_i, o_j \rangle \) and subscription \( s \). They are computed by Equations 6 and 7.

\[
\text{TR}(\langle o_i, o_j \rangle, s) = \frac{o_i, \psi \cap o_j, \psi \cap s, \psi}{(o_i, \psi \cap o_j, \psi) \cup s, \psi} \quad (6)
\]

\[
\text{SR}(\langle o_i, o_j \rangle, s) = 1 - \frac{\text{dist}(o_i, \rho, s, \rho)}{2 \times \text{dist}_{\text{max}}} - \frac{\text{dist}(o_j, \rho, s, \rho)}{2 \times \text{dist}_{\text{max}}}. \quad (7)
\]

Next, we define our Geo-Textual Object Pair Matching (GTOPM) problem.

Definition 5 (GTOPM Problem): Given a geo-textual object pair \( \langle o_i, o_j \rangle \) and a collection of subscriptions \( S \),
III. SOLUTIONS TO OGT-JOIN

This section presents our solution to OGT-Join.

A. BASELINE SOLUTION

A straightforward solution to the OGT-Join problem is presented as follows: Given a sliding window $W$ and geo-textual objects in $W$, denoted by $W(D)$, and a similarity threshold $\theta$, we evaluate each pair of objects $\langle o_i, o_j \rangle$ where $o_i, o_j \in W(D)$. Specifically, we calculate the aggregated similarity between $o_i$ and $o_j$. If $AS(o_i, o_j) \geq \theta$, we add $\langle o_i, o_j \rangle$ into the result set. When the sliding window moves forward by one step, we evaluate each pair of objects in the updated sliding window.

**Theorem 6:** Let $N$ be the number of geo-textual objects we processed from the data stream and $|W|$ be the number of objects in sliding window $W$. Assume that $N \gg |W|$. The time complexity of the baseline solution to the OGT-Join problem is $O(|W|^2 \cdot N)$.

**Proof:** The baseline algorithm requires $O(|W|^2)$ to solve the join problem for objects in $W$. The sliding window moves forward $N$ step. Each time $W$ moves forward, we need $O(|W|^2)$ to re-solve the join problem. As such, the overall time complexity is $O(|W|^2 \cdot N)$.

B. INCREMENTAL ONLINE JOIN ALGORITHM

Based on Theorem 6, we see that the baseline solution is very time consuming. Hence, we need to develop a more efficient solution to the OGT-Join problem. We observe that it is unnecessary to re-solve the join problem each time the sliding window moves forward. In particular, when sliding window $W$ moves forward by one step, the most up-to-date geo-textual object, denoted by $o_e$, is inserted into $W$, while the earliest object, denoted by $o_c$, is removed from $W$. As a result, we only need to consider object pairs in $W$ that contain $o_c$ or $o_e$.

Algorithm 1 presents the pseudo code of our Incremental Online Join (IOJ) algorithm. The inputs of IOJ are the new object $o_e$, sliding window $W$, similarity threshold $\theta$, and the current join result set $R$. The output is the updated join result set $R'$. At the beginning, we initialize $R'$ as $R$ and set $o_e$ to be the earliest object in $W$ (Lines 1–2). For each object pair $p$ in the current join result set $R$, we check if the earliest object $o_e$ is included by $p$. If $o_e \in p$, we remove $p$ from the updated set $R'$ (Lines 3–5). Next, we evaluate all new object pairs related to the new object $o_e$. Specifically, for each object pair related to $o_e$ (i.e., $\langle o_i, o_e \rangle$) we calculate the aggregated similarity between $o_i$ and $o_e$ (i.e., $AS(o_i, o_e)$). If $AS(o_i, o_e) \geq \theta$, we add object pair $\langle o_i, o_e \rangle$ to $R'$ (Lines 6–9). Then, we update the sliding window $W$ by removing the earliest object $o_c$ and inserting the new object $o_e$ from the data stream (Lines 10–11). Finally, we return $R'$, the updated join result set, as the output. **Theorem 7** presents the time complexity of our IOJ algorithm.

**Theorem 7:** Let $N$ be the number of geo-textual objects we processed from the data stream and $|W|$ be the number of objects in sliding window $W$. The time complexity of the baseline solution to the OGT-Join problem is $O(|W|^2 + |W| \cdot N)$.

**Proof:** At the beginning, the IOJ algorithm requires $O(|W|^2)$ to solve the join problem for objects in $W$. When the sliding window moves forward 1 step, we need $O(|W|)$ to evaluate the object pairs with the new object. Thus, the overall time complexity is $O(|W|^2 + |W| \cdot N)$.

IV. SOLUTIONS TO THE GTOPM PROBLEM

We present our generic framework and algorithms to solve the GTOPM problem. Recall that we targets two types of location-aware subscriptions: RLS and SLS. As such, our proposed framework and algorithms should capable of handling both RLS and SLS.

A. GENERIC LOCATION-AWARE SUBSCRIPTION INDEX

Figure 2 presents the framework of our generic location-aware subscription index. The subscription pool consists of three RLS (i.e., $s_1, s_2, s_4$) and three SLS (i.e., $s_5, s_3, s_5$). To index these subscriptions, we employ a grid structure. In Figure 2,
Algorithm 2 OPMR

**Input:** New object pair \( \langle o_i, o_j \rangle \), Subscription index \( I \)
**Output:** A set of subscriptions that match \( \langle o_i, o_j \rangle \)

1. \( c_i \leftarrow \text{cell that covers } o_i \);  
2. \( c_j \leftarrow \text{cell that covers } o_j \);  
3. \( S(c_i) \leftarrow \text{subscriptions indexed by } c_i \);  
4. \( S(c_j) \leftarrow \text{subscriptions indexed by } c_j \);  
5. \( C \leftarrow S(c_i) \cap S(c_j) \);  
6. For each \( s \in C \) do  
   7. if \( s \psi \) matches \( o_i, \psi \cap o_j, \psi \) then  
      8. \( R.add(s) \);  
9. return \( R \);  

and the subscription index \( I \). The output \( R \) is the subscriptions that match the new object pair. At the beginning, we initialize our result set \( R \) to be empty set (Line 1). Next, we evaluate each cell in the grid index and check if the subscriptions indexed by the cell can be pruned. Specifically, for each cell \( c_k \) in \( I \), we first check if the maximum spatial relevance score between pair \( \langle o_i, o_j \rangle \) and \( c_k \) is no less than the minimum value of \( \phi \) for all subscriptions indexed under \( c_k \) (Line 3). If the condition is satisfied, we proceed to run individual check for each subscription indexed under \( c_k \) (Lines 4–6). Otherwise, we filter out all subscriptions under \( c_k \) and proceed to the next cell. Finally, we return \( R \) as the result (Line 7).

V. EXPERIMENTS

We report on experiments with on two real-life datasets. We evaluate the following three methods.

- **Baseline** – The baseline algorithm is presented in Section III-A. It applies baseline solution to the OGT-Join and applies straightforward method to solve the object pair matching problem.
- **OMPR** – OMPR algorithm is presented in Algorithm 2. Note that IOJ (Algorithm 1) is applied for solving the OGT-Join problem.
- **OMPS** – OMPS algorithm is presented in Algorithm 3. Note that IOJ (Algorithm 1) is applied for solving the OGT-Join problem.  

Note that OMPR and OMPS target different types of location-aware subscriptions (i.e., RLS and SLS). So we evaluate OMPR and OMPS separately.

A. DATASETS AND SUBSCRIPTION GENERATION

Our experiments are conducted on two datasets: YEL and TWI. YEL is a geo-textual dataset collected from Yelp, which consists of worldwide business comments with geographical information and textual information. The dataset TWI is a large dataset crawled from Twitter, comprising geo-tagged tweets with locations.

1) Generating SLS

To generate SLS, we randomly select a particular proportion of geo-textual objects each dataset. The locations of selected geo-textual objects are considered to be subscription...
locations. For query keywords, we randomly select a particular number of terms from the textual description of each selected object.

2) GENERATING RLS
The RLS subscription pool is generated as follows. Similarly, we randomly select a particular proportion of geo-textual objects each dataset. The locations of selected geo-textual objects are considered to be the centers of subscription regions. For keyword-based subscription predicates, we randomly select a particular number of terms from the textual description of each selected object and connect them by AND or OR semantics.

3) GENERATING GEO-TEXTUAL DATA STREAMS
For dataset YEL, we regard each business comment as a geo-textual object. For TWI, each geo-tagged tweet is considered to be a geo-textual object. The arrival rate (throughput) is set based on specific experimental settings, which will be presented in experimental results.

B. EXPERIMENTAL RESULTS FOR PROCESSING SLS
We present the experimental settings and results for processing SLS. Parameter settings are presented in Table 1.

| TABLE 1. Parameter settings and default values. |
|-----------------------------------------------|
| relevance threshold (φ) | YEL | TWI |
| default: 0.8 | 0.5–0.9 | 0.5–0.9 |
| preference parameter (α_r) | default: 0.5 | 0.1–0.9 |
| default: 600K | default: 1M |
| number of subscriptions | 200K–1M | 1M–4M |

1) EFFECT OF SIMULATION TIME
In this set of experiments, both methods run for 30 minutes. For Figures 3(a) and 3(b), we generate the geo-textual data stream with the arrival rate of 1/s. For Figure 3(c), we generate the geo-textual data stream with the highest arrival rate that can be handled by each method. We record the average runtime for processing an geo-textual object (i.e., the sum of runtimes for processing geo-textual object join and object matching). From Figures 3(a) and 3(b) we see that OMPS performs consistently better than the baseline on both datasets. Additionally, the results shown in Figure 3(c) suggest that OMPS is capable of handling a stream with much higher throughput, demonstrating the effectiveness of our IOJ and object matching algorithms.

2) EFFECT OF RELEVANCE THRESHOLD φ
This set of experiments evaluate the performance of both methods when varying the relevance threshold defined by each subscription. As shown in Figure 4, both methods perform better when we increase the relevance threshold. The reason is that when the value of φ is increased, the number of object pairs that can match each subscription may reduce. In addition, we find that OMPS exhibits a more significant runtime decreasing trend compared with the baseline. This phenomenon is resulted from the pruning effect of OMPS.

3) EFFECT OF PREFERENCE PARAMETER α_r
We proceed to evaluate the effect of preference parameter α. We vary α_r from 0.1 to 0.9. A higher value of α_r indicates more emphasis on spatial proximity compared against text relevancy. From Figure 5, we see that the baseline method performs consistently as we vary the value of α_r. In contrast, OMPS performs better when we increase α_r. The reason is that OMPS has strong spatial pruning power (i.e., filtering techniques based on the spatial subscription index).

4) EFFECT OF THE NUMBER OF SUBSCRIPTIONS
This set of experiments evaluate the scalability of each method for processing SLS. From Figure 6 we see that the runtime of both methods increases linearly as we increase the number of subscriptions. We also find that our proposed OMPS performs consistently better than the baseline as we vary the number of subscriptions. As such, OMPS is capable of supporting more subscriptions over the geo-textual data stream.
Y. Zhong et al.: Pairwise Location-Aware Publish/Subscribe for Geo-Textual Data Streams

C. EXPERIMENTAL RESULTS FOR PROCESSING RLS

We proceed to present our experimental settings and results for processing RLS. Parameter settings are presented in Table 2.

| Parameter settings and default values. |
|----------------------------------------|
|                                          |
| subscription radius (φ)                |
| default: random                        |
| number of subscriptions                |
| default: 600K                          |

1) EFFECT OF SIMULATION TIME

In this set of experiments, both methods run for 30 minutes. Similarly, for Figures 7(a) and 7(b), we generate the geo-textual data stream with the arrival rate of 1/s. For Figure 7(c), we generate the geo-textual data stream with the highest arrival rate that can be handled by each method. We record the average runtime for processing an geo-textual object (i.e., the sum of runtimes for processing geo-textual object join and object matching). From Figures 7(a) and 7(b) we see that OMPR performs consistently better than the baseline on both datasets. The results shown in Figure 7(c) suggest that OMPR is capable of handling a stream with much higher throughput, demonstrating the effectiveness of our IOJ and object matching algorithms.

2) EFFECT OF SUBSCRIPTION RADIUS

Next, we evaluate the performance as we vary the subscription radius. From Figure 8 we see that both methods perform worse as we enlarge the subscription region. The reason is that enlarging the subscription region may increase the likelihood of a subscription being matched by a new object pair. As such, group filtering (i.e., cell-aware filtering) may be less likely to be triggered.

3) EFFECT OF THE NUMBER OF SUBSCRIPTION

Finally, we evaluate the scalability performance of baseline and OMPR. From Figure 9 we see that OMPR scales well as we increase the number of subscriptions from 200K to 1M and from 1M to 4M on YEL and TWI, respectively. In contrast, the baseline method performs worse in terms of scalability.
VI. RELATED WORK

A. GEO-TEXTUAL OBJECT PUBLISH/SUBSCRIBE

Geo-textual object publish/subscribe systems have been studied extensively in recent years. Given a stream of geo-textual object, for each new object, the system checks whether the spatial and textual information on the new object matches each subscription. Based on the matching conditions, existing studies can be classified into two categories:

1. Predicate-aware matching condition: In this category, a circular or rectangular region is regarded as the subscription spatial predicate [3], [40], [44]. Specifically, if the location of a new geo-textual object falls in the subscription region, spatial matching condition is met.

2. Similarity-aware matching condition: In this category, a similarity score is considered to be the matching condition [4], [6], [14], [15], [48]. In particular, existing studies in this category have two types of matching condition: spatial matching condition and textual matching condition. For spatial matching condition, its spatial similarity score is calculated by the spatial proximity between new object and subscription location. Some studies regard the overlap area between a subscription and a region of geo-textual objects as the spatial similarity score [7], [12], [45]. The textual matching condition is regarded as the textual relevance between new object and subscription keywords. Some recent studies use term frequency as the matching condition [1], [8], [11], [21], [22], [33], [42]. Specifically, top-\(k\) frequent terms or terms with frequency no less than a pre-defined threshold are delivered to users.

These studies only consider single object in their matching condition, which cannot be applied to our object-pair based publish/subscribe.

B. SPATIAL KEYWORD SEARCH

Spatial keyword search is a fundamental research topic over the past decade. Mokbel and Magdy [23], Magdy et al. [20], and Chen et al. [10] provide comprehensive studies on general research problems regarding spatial keyword search. Given a collection of geo-textual objects, spatial keyword search aims to process spatial keyword queries over the geo-textual dataset. Here, spatial keyword queries are standalone query and they contain both spatial and textual requirements. Spatial requirement can be defined by the spatial proximity between query and object. Textual requirement can be formulated as keyword predicates or text relevancy.
Besides traditional spatial keyword queries, advanced spatial keyword queries with more requirements were investigated in recent years, which include group-based spatial keyword queries [16]–[18], sequence matching [38], route and trajectory based queries [2], [9], [13], [25], [30]–[32], [41], and advanced trajectory search [24], [26]–[29], [49]. Apart from the aforementioned spatial keyword query processing studies, some reinforcement learning based methods [35], [36], [43], distance metric optimization [34], classification of imbalanced data [37], location-based recommender systems [19], [39], [46] can be applied to further enhance the efficiency and effectiveness of spatial keyword search.

VII. CONCLUSION
We studied the problem of developing location-aware publish/subscribe system that feeds subscribers with geo-textual object pairs. To address the problem, we proposed a two-phase subscription matching scheme, which includes online geo-textual join and object pair matching. Our proposal is capable of handing two types of popular subscription queries: region-based subscription and similarity-based subscription. Our experimental study suggested that our proposal was able to achieving high efficiency and effectiveness in comparison with baseline.

REFERENCES
[1] P. Ahmed, M. Hasan, A. Kashyap, V. Hristidis, and V. J. Tsotras, “Efficient computation of top-k frequent terms over spatio-temporal ranges,” in Proc. SIGMOD, 2017, pp. 1227–1241.
[2] X. Cao, L. Chen, G. Cong, and X. Xiao, “Keyword-aware optimal route search,” in Proc. VLDB, vol. 5, no. 11, pp. 1136–1147, Jul. 2012.
[3] L. Chen, G. Cong, and X. Cao, “An efficient query indexing mechanism for filtering geo-textual data,” in Proc. SIGMOD, 2013, pp. 749–760.
[4] L. Chen, G. Cong, X. Cao, and K. Tan, “Temporal spatial-keyword top-k publish/subscribe,” in Proc. ICDE, 2015, pp. 255–266.
[5] L. Chen, G. Cong, C. S. Jensen, and D. Wu, “Spatial keyword query processing: An experimental evaluation,” in Proc. VLDB, vol. 6, no. 3, pp. 217–228, 2013.
[6] L. Chen and S. Shang, “Approximate spatio-temporal top-k publish/subscribe,” World Wide Web, vol. 22, no. 5, pp. 2153–2175, Sep. 2019.
[7] L. Chen and S. Shang, “Region-based message exploration over spatio-temporal data streams,” in Proc. AAAI, 2019, pp. 873–880.
[8] L. Chen, S. Shang, C. S. Jensen, J. Xu, P. Kalnis, B. Yao, and L. Shao, “Top-k term publish/subscribe for geo-textual data streams,” VLDB J., vol. 29, no. 5, pp. 1101–1128, 2020.
[9] L. Chen, S. Shang, C. S. Jensen, B. Yao, Z. Zhang, and L. Shao, “Effective and efficient reuse of past travel behavior for route recommendation,” in Proc. KDD, 2019, pp. 488–498.
[10] L. Chen, S. Shang, C. Yang, and J. Li, “Spatial keyword search: A survey,” Geoinformatica, vol. 24, no. 1, pp. 85–106, Jan. 2020.
[11] L. Chen, S. Shang, Z. Zhang, X. Cao, C. S. Jensen, and P. Kalnis, “Location-aware top-k term publish/subscribe,” in Proc. ICDE, 2018, pp. 749–760.
[12] L. Chen, S. Shang, K. Zheng, and P. Kalnis, “Cluster-based subscription matching for geo-textual data streams,” in Proc. ICDE, 2019, pp. 890–901.
[13] D. Guo, Y. Zhu, W. Xu, S. Shang, and Z. Ding, “How to find appropriate automobile exhibition halls: Towards a personalized recommendation service for auto shows,” Neurocomputing, vol. 213, pp. 95–101, Nov. 2016.
[14] H. Yu, Y. Liu, G. Li, J. Feng, and K. Tan, “A location-aware publish/subscribe framework for parameterized spatio-temporal subscriptions,” in Proc. ICDE, 2015, pp. 711–722.
[15] J. Hu, R. Cheng, D. Wu, and B. Jin, “Efficient top-k subscription matching for location-aware publish/subscribe,” in Proc. SSTD, 2015, pp. 333–351.
[16] N. M. Kou, Y. Li, H. Wang, L. H. U, and Z. Gong, “Crowdsourced top-k queries by confidence-aware pairwise judgments,” in Proc. SIGMOD, 2017, pp. 1415–1430.
[43] S. Yang, Y. Gao, B. An, H. Wang, and X. Chen, “Efficient average reward reinforcement learning using constant shifting values,” in Proc. AAAI, 2016, pp. 2258–2264.

[44] M. Yu, G. Li, and J. Feng, “A cost-based method for location-aware publish/subscribe services,” in Proc. CIKM, 2015, pp. 693–702.

[45] M. Yu, G. Li, T. Wang, J. Feng, and Z. Gong, “Efficient filtering algorithms for location-aware Publish/Subscribe,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 4, pp. 950–963, Apr. 2015.

[46] Y. Yu, H. Wang, S. Sun, and Y. Gao, “Exploiting location significance and user authority for point-of-interest recommendation,” in Proc. PAKDD, 2017, pp. 119–130.

[47] K. Zhao, L. Chen, and G. Cong, “Topic exploration in spatio-temporal document collections,” in Proc. SIGMOD, 2016, pp. 985–998.

[48] K. Zhao, Y. Liu, Q. Yuan, L. Chen, Z. Chen, and G. Cong, “Towards personalized maps: Mining user preferences from geo-textual data,” Proc. VLDB, vol. 9, no. 13, pp. 1545–1548, 2016.

[49] Y. Zhao, S. Shang, Y. Wang, B. Zheng, Q. V. H. Nguyen, and K. Zheng, “REST: A reference-based framework for spatio-temporal trajectory compression,” in Proc. KDD, Jul. 2018, pp. 2797–2806.

YING ZHONG is currently an Associate Professor with the School of Computer and Information Engineering, Xiamen University of Technology. Her research interests include big data management and analytics, and information retrieval.

SHUNZHI ZHU received the Ph.D. degree from Xiamen University, in 2007. He was a Visiting Professor with Florida International University, USA. He is currently a Full Professor with the School of Computer and Information Engineering, Xiamen University of Technology. His research interests include big data management and analytics, data mining, and information retrieval.

YAN WANG received the Ph.D. degree from Renmin University, in 2010. He is currently an Associate Professor with the School of Computer and Information Engineering, Xiamen University of Technology. His research interests include big data, data mining, and smart city.

JIANMIN LI received the Ph.D. degree from Xiamen University, in 2015. He is currently an Associate Professor with the School of Computer and Information Engineering, Xiamen University of Technology. His research interests include big data, machine learning, and pattern recognition.

XINXIN ZHANG received the Ph.D. degree from the Institute of Remote Sensing, Chinese Academy of Sciences, in 2011. He is currently an Associate Professor with the School of Computer and Information Engineering, Xiamen University of Technology. His research interests include remote sensing and pattern recognition.

JEDI S. SHANG is currently a Principle Scientist with Thinvent Digital Technology Company Ltd. His research interests include smart city and intelligent transportation.

VOLUME 8, 2020