Coverage and Diversity Aware Top-k Query for Spatio-Temporal Posts

Paras Mehta
Freie Universität Berlin
Germany
paras.mehta@fu-berlin.de

Dimitrios Skoutas
IMIS, Athena R.C.
Greece
dskoutas@imis.athena-innovation.gr

Dimitris Sacharidis
Technische Universität Wien
Austria
dimitris@ec.tuwien.ac.at

Agnès Voisard
Freie Universität Berlin
Germany
agnes.voisard@fu-berlin.de

ABSTRACT

Large amounts of user-generated content are posted daily on the Web, including textual, spatial and temporal information. Exploiting this content to detect, analyze and monitor events and topics that have a potentially large span in space and time requires efficient retrieval and ranking based on criteria including all three dimensions. In this paper, we introduce a novel type of spatial-temporal-keyword query that combines keyword search with the task of maximizing the spatio-temporal coverage and diversity of the returned top-k results. We first describe a baseline algorithm based on related search results diversification problems. Then, we develop an efficient approach which exploits a hybrid spatial-temporal-keyword index to drastically reduce query execution time. To that end, we extend two state-of-the-art indices for top-k spatio-textual queries and describe how our proposed approach can be applied on top of them. We evaluate the efficiency of our algorithms by conducting experiments on two large, real-world datasets containing geotagged tweets and photos.

CCS Concepts

•Information systems → Spatial-temporal systems; Information retrieval diversity;

Keywords

spatio-temporal keyword queries, coverage, diversity

1. INTRODUCTION

With the widespread use of online social networks and GPS-enabled mobile devices, there are large amounts of content created daily containing textual, spatial and temporal information. Typical examples include geotagged tweets, photos or check-ins, where the textual content is a short text or a set of tags, while the spatio-temporal content refers to the location and time of the post. Analyzing such data is valuable in a wide range of applications, such as event detection [23, 20], topic detection [10] and opinion mining [26]. Moreover, users often want to browse and navigate across content in microblogs to track and monitor the evolution of events and stories as they unfold in the dimensions of space and time. A basic functionality for any such analysis and exploratory search is spatio-temporal keyword queries, i.e. queries including filters in each of these three dimensions: text, space and time. Over the past years, both spatial keyword queries and temporal information retrieval have been studied extensively, exploring and combining techniques at the intersection of these fields [12, 9, 5, 3].

With respect to spatial keyword queries, several variants have been studied, depending on the type of textual and spatial predicates used. The text part comprises a set of keywords, which can be used either for ranked retrieval, e.g. ranking documents or web pages based on term frequencies, or as boolean filters, e.g. when searching through short text messages or metadata. Similarly, the spatial part may specify a location, in which case the results can be ranked by proximity to it, or a spatial region, which acts as a boolean filter. For instance, the survey presented in [9] identifies the following types of queries: *Boolean Range Query*, which applies a set of keywords and a spatial region as boolean filters, returning all matching documents; *Boolean kNN Query*, which applies a set of keywords as boolean filter, and ranks the results based on their proximity to the query location, returning the k nearest neighbors; and *Top-k kNN Query*, which retrieves the top-k documents based on an aggregate score combining both text relevance and spatial proximity. Similarly, temporal information retrieval combines the notion of text relevance with temporal relevance [5, 3]. As with space, the temporal condition may specify either a single point in time, e.g. when ranking documents by recency, or a time interval, e.g. for retrieving all posts within a given time window.

Nevertheless, the problem becomes even more complex and challenging when all three dimensions need to be taken

SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA

ISBN 978-1-4503-4589-7/16/10...$15.00

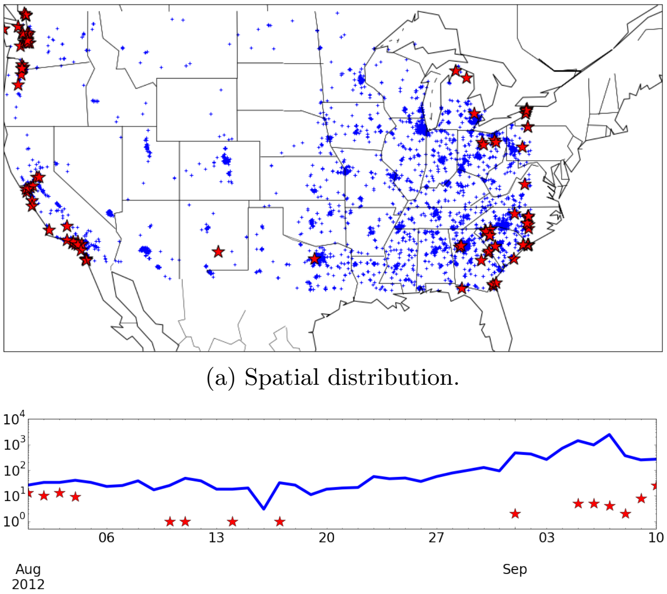
DOI: http://dx.doi.org/10.1145/2996913.2996941

into account during the retrieval and selection of results. Consider a user searching microblogs for information about a topic or event. For example, the blue dots/lines in Figures 1(a) and 1(b) depict, respectively, the spatial and temporal distribution of tweets in the U.S. for a search with keywords “obama, election”, for a period of 40 days starting on 01/08/2012. This search returns thousands of results. Ranking results by textual relevance is often not suitable when it involves short texts or tags – essentially, every post that contains the query keyword(s) is relevant. Instead, selection and ranking of relevant posts based on their spatial and temporal attributes is much more interesting. As shown from the aforementioned query types, ranking on these dimensions typically assumes that a single point in time and space is specified, so that the posts can be ranked according to their proximity to it. However, for topics or events that have a long span in space and time, as in this example, there is not a single “central” point to use for spatio-temporal ranking. Thus, there is a need for a query type that allows for specifying a desired spatial range and time window, while still being able to retrieve top-k results according to spatio-temporal criteria.

To that end, we introduce in this paper a novel type of query, the *top-k Coverage and Diversity aware Spatio-Temporal Keyword* (kCD-STK) query. Intuitively, the goal is to return top-k results, where the ranking is driven by the spatio-temporal distribution of the posts. Thus, we consider as more relevant, posts that lie within dense areas in the three-dimensional spatio-temporal space. Specifically, we introduce the criterion of *spatio-temporal coverage*, which assigns a score to each post based on the number of other posts that lie within a specified distance threshold to it in the spatio-temporal dimensions. Furthermore, to avoid over-representing these areas while missing other interesting results, we also try to maximize the spatio-temporal *diversity* among the selected posts. Returning to the example presented in Figure 1, the red stars correspond to a subset of 100 results selected by the kCD-STK query. Notice that the selected results are more spread out in space and time, instead of focusing around a single area, thus better representing the whole set of relevant posts.

The kCD-STK query is founded on the basic concepts commonly used for search results diversification. In particular, it introduces the concept of coverage [15] in the search results diversification framework presented in [18] (see Section 2.2 for more details). By determining the relevance of each result to the query indirectly, i.e. based on the number of other results it covers, it allows the spatio-temporal filters in the query to be defined more flexibly, indicating a whole spatial region and a time window rather than requiring the user to restrict his search around a specific location and point in time. This makes the kCD-STK query more suitable for exploratory search. As the returned top-k results more closely reflect the spatio-temporal distribution of the whole result set, they can serve as “anchor” points for further exploration of the available posts.

Since typical diversification problems are known to be NP-hard, the challenge that arises in practice is how to efficiently evaluate a kCD-STK query, so that the results can still be retrieved in real time. The aforementioned approaches are general frameworks for results diversification, thus none of them deals particularly with the spatio-temporal coverage or diversity of posts. To the best of our knowledge, our work is the first to introduce these criteria and to consider their efficient evaluation in the context of spatial-temporal-keyword queries. More specifically, the main contributions of our work can be summarized as follows:



(a) Spatial distribution.

(b) Temporal distribution.

Figure 1: Example of results returned by a boolean query (blue) and the corresponding kCD-STK query (red).

- We formally introduce a novel type of spatial-temporal-keyword query, the kCD-STK query. This query allows a keyword search to be issued with spatial and temporal range filters, and then ranks the matching results according to the criteria of *spatio-temporal coverage* and *diversity*.
- We propose an efficient strategy for evaluating a kCD-STK query. Then, we show how state-of-the-art hybrid spatio-textual indices can be adapted and extended to be used with this strategy for efficiently selecting the top-k results from the whole set of relevant posts.
- We experimentally evaluate our approach, using two large, real world datasets containing geotagged tweets and photos. The experiments demonstrate that our approach can effectively exploit the underlying index structure, thus significantly reducing the time for computing the top-k coverage and diversity aware results.

The rest of the paper is structured as follows. Section 2 reviews related work, focusing on spatial-temporal-keyword queries and search results diversification. Then, the kCD-STK query is formally introduced in Section 3, defining the criteria for spatio-temporal coverage and diversity. Section 4 presents our approach and describes how it can be applied with state-of-the-art hybrid indices for spatial keyword queries, after extending them to include the temporal dimension. Finally, Section 5 presents our experimental evaluation, and Section 6 concludes the paper.

2. RELATED WORK

Next, we review related work on spatial and temporal keyword queries and on search results diversification.

2.1 Spatial and Temporal Keyword Queries

Spatial keyword queries have received a lot of attention over the past years. The main focus is on combining spatial and textual indices into hybrid ones, and investigating different query evaluation strategies (e.g. text-first vs. space-first [12]). In building hybrid index structures, various combinations have been proposed, including the use of an R-tree, grid or space filling curve for the spatial part and an inverted file or bitmap for the textual part. A comprehensive survey and comparison of existing approaches can be found in [9]. Characteristic examples include the IF-R*-tree, where the top-level index is an inverted file with the postings in each inverted list indexed by an R-tree, and the R*-tree-IF, where the top-level index is an R*-tree with inverted files attached to each leaf node [32]. Several similar variants exist (e.g. the IR-tree [29]), while other works have combined inverted files with grid [25] or space filling curves [11, 12].

Two state-of-the-art approaches for top-k spatial keyword queries are the I^3 hybrid index [31] and the RCA algorithm [30]. The I^3 index maintains a Quadtree for each keyword, indexing the documents containing it. Each keyword is used as a key in a lookup table and it is associated with a pointer. If the documents containing this keyword can fit in a single disk page, the pointer links directly to that page; otherwise, it points to the root of a Quadtree which spatially indexes the relevant documents. The leaf nodes of the Quadtree point to the disk pages where the documents are stored.

The RCA approach uses only an inverted index. In particular, it maintains two inverted lists for each keyword. The first is a standard inverted list which stores the documents containing the keyword in decreasing order of relevance. The second stores documents according to the Z-order encoding of their coordinates [17]. Query processing exploits the following property of the Z-order encoding. Assume a spatial bounding box R, with z_{min} and z_{max} being the Z-order encodings of its top-left and bottom-right corners, respectively. Then, the Z-order encoding of any point that lies within R has a value $z \in [z_{min}, z_{max}]$. This allows to efficiently process top-k queries using an adaptation of the CA algorithm [16] for rank aggregation.

All aforementioned approaches consider only the spatial dimension of documents. On the other hand, a large amount of research in temporal information retrieval (TIR) exists as well (see [5, 3] for recent surveys). However, only few works have considered both dimensions of space and time in keyword queries. The index presented in [21] comprises a shallow R-tree extended with an inverted index at each leaf node to index the terms of the contained documents. To deal with the temporal dimension, the original document ids are replaced with new ones that are assigned chronologically, thus facilitating the retrieval of documents within a given temporal range.

In a different direction, the problem of continuously maintaining top-k most relevant results over a stream of geotextual documents is presented in [8]. This work combines the criteria of text relevance, spatial proximity, and recency. Finally, other works in TIR have dealt with timelines and summaries of event-related information in microblogs [2, 19].

However, these approaches either apply the spatio-temporal criteria as boolean filters or use them to rank documents based on spatial proximity and/or recency. To the best of our knowledge, our work first introduces the criteria of spatio-temporal coverage and diversity in keyword queries.

2.2 Search Results Diversification

Ranking search results purely by relevance often leads to including many similar documents in the top results, hence causing repetition and redundancy in the result set. Search results diversification has been proposed as a more advanced technique for selecting a subset of results to present to the user. The goal is to improve the utility of the results by increasing their novelty, thus improving the user experience, especially during exploratory search. More specifically, content-based diversification aims at selecting a subset of documents that maximizes an objective function with two components: *relevance* and *diversity*. The former measures how relevant a result is for the query, while the latter measures the dissimilarity or novelty of that result w.r.t. others already selected.

Many different formulations have been proposed for search results diversification (refer to [18, 14] for classification). The most well-known approach is the framework proposed in [18]. According to it, the problem is defined as selecting a subset $\mathcal{R}^*$ of the whole result set $\mathcal{R}$, with $|\mathcal{R}^*| = k$, that maximizes an objective function ϕ, which combines the criteria of relevance and diversity. There exist different ways to define ϕ, leading to different variants of the problem. For example, in the MaxSum variant, ϕ is defined as the weighted sum of two components: the total relevance of documents and the sum of pairwise distances among the documents.

As shown in [18], the MaxSum problem, as well as other similar variants, is NP-hard by reduction to the MaxSumDispersion problem. Thus, greedy heuristics are used in practice to efficiently compute a diversified subset of the results. The main approach is to incrementally construct the diversified result set by choosing at each step the object that maximizes a certain scoring function. A well-known function for this purpose is the *maximal marginal relevance* (*mmr*) [6]. An evaluation of various object scoring functions and different heuristics can be found in [28].

Other types of diversification problems have also been studied, such as taxonomy/classification-based diversification [1, 27] or multi-criteria diversification [13]. Closer to our work is the coverage problem [15]. Here, the goal is to select the minimum subset of documents such that the selected documents are diverse, i.e. have distance to each other at least ϵ, and *cover* the whole dataset, i.e. each remaining object lies within distance ϵ from a selected one. This formulation is suitable for data exploration and summarization; however, in this case the size of the selected subset is not fixed, but depends instead on the distance threshold ϵ.

More recently, diversity has also been considered in the context of publish/subscribe systems for text streams [7]. In this setting, the top-k results are continuously maintained over a stream of documents. A newly arriving document enters the top-k results if: (a) it contains any of the query keywords and (b) replacing the oldest document in the current result set with the new one increases the overall relevance and diversity of the results.

In our work, we combine the criterion of coverage from [15] with the general diversification framework of [18]. Thus, the relevance of each result is determined indirectly based on the number of other results it covers from the original set, while the number of results to return is still explicitly specified in the query. Moreover, all aforementioned works focus on formulating the diversification problem in a generic manner, using abstract definitions for document relevance and distance. Subsequently, the efficiency of computation is addressed by introducing heuristic algorithms that compute an approximation of the optimal solution. In our work, we focus on the specific problem of selecting spatio-temporally diverse subsets of results. We define concrete criteria for spatio-temporal coverage and diversity, and we show how an underlying spatio-temporal index can be exploited to further speed up the computation.

3. PROBLEM DEFINITION

We first provide the basic definitions necessary to formulate the problem at hand.

Definition 1 (Post). *A spatio-temporal post D is represented by a tuple $D = \langle loc, t, \Psi \rangle$, where $loc = (x, y)$ are the coordinates of the location where the post was made, t is the timestamp of the post, and Ψ is a keyword vector containing zero or more terms, keywords or tags contained in the post.*

Definition 2 (STK Filter). *A spatial-temporal-keyword filter F is a tuple $F = \langle R, T, \Psi \rangle$, where the spatial filter $R = [(x_{min}, y_{min}), (x_{max}, y_{max})]$ specifies a spatial bounding box, the temporal filter $T = [t_{min}, t_{max}]$ specifies a time window, and the keyword filter $\Psi = \{\psi_1, \psi_2, \ldots, \psi_n\}$ specifies a set of keywords.*

For the remainder of this paper, we use dot notation to refer to a tuple's attribute values. The next definition determines when a post is considered relevant for a given STK filter.

Definition 3 (Relevant Posts). *Given a collection $\mathcal{D}$ of posts and an STK filter F, the set of relevant posts $\mathcal{D}_F$ contains all posts $D \in \mathcal{D}$ such that (i) $D.loc \in F.R$, (ii) $D.t \in F.T$, and either (iii-a) $D.\Psi \cap F.\Psi \neq \emptyset$ under OR semantics, or (iii-b) $D.\Psi \supseteq F.\Psi$ under AND semantics.*

Notice that the difference between OR and AND semantics is whether a relevant post must contain *at least one* or *all* of the keywords that appear in the filter.

As discussed in Section 1, for the type of posts and STK filters that motivate our work, i.e. exploratory search for topics or events that are distributed across potentially large intervals in space and time, the number of relevant posts is typically very high. Therefore, our objective is to select a small subset of k relevant posts that have high *coverage* and *diversity*. To elaborate on these two notions, we first need to introduce measures of spatial and temporal distance between two relevant posts (w.r.t. an STK filter F) $D_i, D_j \in \mathcal{D}_F$.

The spatial distance is defined as:

$$d_s(D_i, D_j) = \frac{d(D_i.loc, D_j.loc)}{\sigma_{max}}$$

where $d((x, y), (x', y'))$ is the Euclidean distance between two points and σ_{max} is a normalization factor corresponding to the length of the diagonal of $F.R$, i.e. the maximum possible spatial distance between any pair of posts lying in $F.R$. Note that it is possible to use other functions (e.g. L_p norms) to measure spatial distance; the changes to our methodology are straightforward.

Similarly, the temporal distance is defined as:

$$d_t(D_i, D_j) = \frac{|D_i.t - D_j.t|}{\tau_{max}}$$

where $\tau_{max} = F.t_{max} - F.t_{min}$ is a normalization factor corresponding to the maximum possible temporal distance. As before, one could also employ other functions for the temporal distance, e.g. to assign greater importance to more recent posts.

We are now ready to introduce our two key notions, *coverage* and *diversity*. We first define them for individual posts, and then extend the definitions to sets of posts.

Definition 4 (Coverage). *Given a collection $\mathcal{D}$ of posts and an STK filter F, the coverage of a post $D \in \mathcal{D}_F$ is:*

$$cov(D) = \frac{1}{|\mathcal{D}_F|} |\{D' \in \mathcal{D}_F : d_s(D, D') \leq \rho_s \;\&\; d_t(D, D') \leq \rho_t\}|, \tag{1}$$

where $\rho_s, \rho_t \in [0, 1]$ are unit-less spatial and temporal distance thresholds, respectively. Moreover, the coverage of a set of posts $\mathcal{R} \subseteq \mathcal{D}_F$ of size k is:

$$cov(\mathcal{R}) = \frac{1}{k} \sum_{D \in \mathcal{R}} cov(D). \tag{2}$$

Since each post in the set $\mathcal{R}$ can potentially cover all $|\mathcal{D}_F|$ relevant posts, the denominators in the above equations ensure that coverage takes values in the $[0, 1]$ range.

Definition 5 (Diversity). *Given a collection $\mathcal{D}$ of posts and an STK filter F, the diversity of a pair of posts $D_i, D_j \in \mathcal{D}_F$ is:*

$$div(D_i, D_j) = w \cdot d_s(D_i, D_j) + (1 - w) \cdot d_t(D_i, D_j), \tag{3}$$

where $w \in [0, 1]$ is an application-specific weight parameter between the spatial and the temporal distances. Moreover, the diversity of a set of posts $\mathcal{R} \subseteq \mathcal{D}_F$ of size k is:

$$div(\mathcal{R}) = \frac{1}{k \cdot (k - 1)} \sum_{D_i, D_j \in \mathcal{R}, i \neq j} div(D_i, D_j). \tag{4}$$

As there are $k \cdot (k - 1)$ ordered pairs of posts in set $\mathcal{R}$, the denominator normalizes diversity in the $[0, 1]$ range.

We can now define the Coverage & Diversity aware top-k STK query.

Definition 6 (kCD-STK Query). *Given a collection $\mathcal{D}$ of posts, a Coverage and Diversity aware top-k STK query specifies an STK filter F and seeks for a result set $\mathcal{R}^*$ of k relevant posts such that:*

$$\mathcal{R}^* = \underset{\mathcal{R} \subseteq \mathcal{D}_F, |\mathcal{R}| = k}{\arg\max} \{(1 - \lambda) \cdot cov(\mathcal{R}) + \lambda \cdot div(\mathcal{R})\}, \tag{5}$$

where $\lambda \in [0, 1]$ is a parameter determining the tradeoff between coverage ($\lambda = 0$) and diversity ($\lambda = 1$).

4. METHODOLOGY

We now present our methodology for evaluating the kCD-STK query. It is split into two phases; we first determine the set of relevant posts, and then construct the result set by identifying k posts with high coverage and diversity. For each phase, we state the objective, outline the proposed approach, and then elaborate on the implementation using state-of-the-art index structures from the literature.

4.1 Finding Relevant Posts

For a given STK filter F, the objective of the first phase is to obtain the posts that satisfy F, assuming OR or AND semantics. Our approach is to employ existing techniques used to retrieve documents based on spatial and textual criteria, and extend them to act as filters and, more importantly, to be able to handle the temporal information. Therefore, we discuss next two distinct implementations, one based on the RCA approach [30], and another using the I^3 index [31].

RCA-based Implementation.

We follow the rationale of the RCA method for ranking documents based on a spatio-textual score. Recall that in this method each keyword is associated with two postings lists, one which sorts documents in descending order of textual relevance, and another which sorts documents according to their Z-order encoding of their locations. For our purposes, the first postings list can be ignored. To facilitate filtering using spatial and temporal predicates, we compute the Z-order over the 3D spatio-temporal space. The filtering property of the Z-order encoding, as described in Section 2.1, still holds, and it is used to eliminate posts that lie outside the given spatio-temporal filters.

In particular, the retrieval of relevant posts proceeds as follows.

- Determine the Z-order range $[z^-, z^+]$ that minimally covers the spatial $F.R$ and temporal $F.T$ ranges specified by the filter F.
- For each keyword ψ in the filter $F.\Psi$, retrieve from the corresponding postings list only those posts with Z-order encoding in the $[z^-, z^+]$ range.
- For each keyword, eliminate false positives, i.e. posts within the $[z^-, z^+]$ range that do not satisfy the spatial $F.R$ and temporal $F.T$ ranges. This is a necessary step given the inherent limitation of Z-order encoding [30].
- Merge the lists with the surviving posts per keyword. For OR semantics, return the union, while for AND semantics, return the intersection of the lists.

I^3-based Implementation.

We employ the I^3 index and the associated methodology presented in [31] for retrieving documents based on a spatio-textual score. As with the case of the RCA-based implementation, we need to extend the underlying index structure to support retrieval using both spatial and temporal criteria. Therefore, instead of having a Quadtree associated with each keyword, we construct an Octree indexing documents in the 3D spatio-temporal space. Then, the retrieval of relevant documents proceeds largely similar to [31].

The algorithm is best understood by considering a single virtual (i.e. non-materizalized) Octree. We say that a keyword is dense for a particular cell, if the number of posts that lie within the cell and contain this keyword exceeds the disk page capacity. With each cell, we associate the set of posts that have a non-dense keyword, and for each dense keyword a signature summarizing the posts with that keyword. A cell has children cells if it has at least one dense keyword.

To find the relevant posts for F, we perform a depth-first traversal of the Octree. A cell is only visited if it overlaps with the spatial $F.R$ and temporal $F.T$ ranges. In addition, a cell is pruned if it can be guaranteed that the subtree rooted at this cell contains no relevant posts. This check differs depending on the keyword filter semantics. For OR semantics, the cell is pruned if the associated set of posts is empty and the union of the signatures for the non-dense keywords among $F.\Psi$ is empty. For AND semantics, the cell is pruned if no associated post is contained in the intersection of the signatures for the non-dense keywords among $F.\Psi$. At the end of this traversal, the set of posts associated with all non-pruned leaf cells constitute the set of relevant posts.

4.2 kCD-STK Query Processing

Processing a kCD-STK query is a computationally hard optimization problem. Indeed, if we set parameters $\lambda = 1$ and $w = 1$ for instance, we seek for a set $\mathcal{R}$ of k posts that maximize the objective function $\sum_{i \neq j \in \mathcal{R}} d(D_i.loc, D_j.loc)$. This is precisely the 2D-MaxSumDispersion problem for which no exact, polynomial time algorithm is known (although it remains open whether 2D-MaxSumDispersion is NP-hard, similar MaxSumDispersion problems are [22]). Therefore, we turn to heuristic algorithms for constructing the result set of a kCD-STK query.

In particular, we adopt the standard greedy method for constructing the set incrementally, where at each step the document that has the highest *marginal gain* on the objective function is added. It is known that such an approach gives a 2-approximation for the general MaxSumDispersion problem [4]. In our context, the objective function for a set of posts $\mathcal{R}$ is:

$$\phi(\mathcal{R}) = (1 - \lambda) \cdot cov(\mathcal{R}) + \lambda \cdot div(\mathcal{R}),$$

and the marginal gain $g(D) \equiv \phi(\mathcal{R} \cup \{D\}) - \phi(\mathcal{R})$ for including $D \in \mathcal{D}_F \setminus \mathcal{R}$ is:

$$g(D) = \frac{1-\lambda}{k} \cdot cov(D) + \frac{\lambda}{k \cdot (k-1)} \sum_{D_i \in R} div(D, D_i). \quad (6)$$

In other words, the marginal gain on the objective function of post D is the weighted sum of its coverage and its diversity to the existing posts in the set $\mathcal{R}$. In what follows, we first describe the straightforward approach of implementing the greedy algorithm, which will serve as our baseline. Then, we introduce a generic index-aware methodology that takes advantage of the underlying index structure in order to speed up the greedy algorithm.

4.2.1 *Baseline Greedy Algorithm*

Once all relevant posts have been identified, the *Baseline Greedy Algorithm*, denoted as BSL, directly implements the greedy heuristic for the MaxSumDispersion problem.

Algorithm 1 shows the pseudocode for BSL. Initially, the set of relevant posts is retrieved (line 1), following the methodology discussed in Section 4.1. Then the result set is built incrementally. At each iteration (lines 3–7), the marginal gain of each post is computed by applying Equation 6 (line 5). The post with the highest marginal gain is selected for insertion in the result set (lines 6–7). The algorithm terminates as soon as k posts have been selected (line 3).

When computing the marginal gains, one thing to notice is that the coverage term remains fixed across all iterations for a particular post D. The reason is that $cov(D)$ depends on the fixed set $\mathcal{D}_F$ of relevant posts, rather than the partial result set. Therefore, we only need to compute this first term once for all posts.

4.2.2 *Index-Aware Greedy Algorithm*

The main drawback of the BSL algorithm is that it computes (or updates) the marginal gain for every relevant post up to k times. When the number of relevant posts $|\mathcal{D}_F|$ is

Algorithm 1: Algorithm BSL

```
Input: document collection $\mathcal{D}$, STK filter $F$, result set size $k$
Output: coverage and diversity aware result set $\mathcal{R}^*$
$\mathcal{D}_F \leftarrow$ FindRelevantDocs($\mathcal{D}, F$)          ▷ Section 4.1
$\mathcal{R}^* \leftarrow \emptyset$
while $|\mathcal{R}^*| < k$ do
    foreach $D \in \mathcal{D}_F$ do
        compute $g(D)$                                ▷ Equation 6
    find document $D^*$ that maximizes $g(D^*)$
    $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \{D^*\}$
```

large, this constitutes a performance bottleneck. It would be desirable to avoid computing the marginal gain for posts that are most likely to not be included in the result set. To achieve this goal, we propose the *Index-Aware Greedy Algorithm*, termed IDX, that takes advantage of the existing index structure to speed up kCD-STK query processing. We first overview IDX without specific assumptions on the index, and later delve into implementation details assuming explicitly an RCA or an I^3 approach. We emphasize that our methodology is generic and can be readily applied over other spatio-textual indices (provided they can be extended to handle temporal information).

The basic idea of IDX is to form groups by clustering relevant posts that have similar spatial and temporal information. Thanks to the inherent spatio-temporal clustering of the underlying index, the groups are constructed with negligible overhead. Then, at each iteration and for each group we compute an upper bound on its marginal gain. Groups that are promising, i.e. have a high upper bound, are examined more closely by looking at their members. On the other hand, at each iteration, unpromising groups can be dismissed, thus avoiding to compute the exact marginal gain of their members.

With each group G, we associate the following information.

- Its cardinality $|G|$.
- Its spatial extent $G.R$, which is a rectangle that minimally bounds the locations of the group's posts.
- Its temporal extent $G.T$, which is a time interval that minimally bounds the timestamps of the group's posts.
- A lower bound $G.cov^-$ on the coverage of any post in the group.
- A set $G.par$ that contains groups which are *partially covered* by G. We say that a post covers another if their spatial and temporal distances are within the spatial and temporal distance thresholds respectively. We say that a group G partially covers another G', if there can exist a post D in the former, and two posts in the latter such that one is covered by D, while the other is not.
- A value $G.div^+$ which is an upper bound on the diversity of any post in the group to all posts in $\mathcal{R}$.

Based on this information, we can compute an upper bound $g(G)^+$ on the marginal gain of any member D in group G as follows:

$$g(G)^+ = \frac{1-\lambda}{k} \cdot \left(G.cov^- + \frac{1}{|\mathcal{D}_F|} \sum_{G' \in G.par} |G'| \right) + \frac{\lambda}{k \cdot (k-1)} G.div^+. \tag{7}$$

We next discuss how to derive the group information. To compute $G.cov^-$ and construct $G.par$, we iterate across the groups, and for each such group G', we compute spatial and temporal bounds:

$$d_s^-(G,G') = \frac{\texttt{mindist}(G.R, G'.R)}{\sigma_{max}} \text{ and } d_s^+(G,G') = \frac{\texttt{maxdist}(G.R, G'.R)}{\sigma_{max}}$$

$$d_t^-(G,G') = \frac{\texttt{mindist}(G.T, G'.T)}{\tau_{max}} \text{ and } d_t^+(G,G') = \frac{\texttt{maxdist}(G.T, G'.T)}{\tau_{max}},$$

where the mindist and maxdist are the standard functions that return the minimum and maximum possible distances respectively between ranges (Euclidean distance for spatial ranges, and absolute value for temporal ranges). Intuitively, these values bound the spatial and temporal distances between any pair of posts from groups G and G'. We thus distinguish the following cases:

- $d_s^+(G,G') \leq \rho_s$ and $d_t^+(G,G') \leq \rho_t$: we increment $G.cov^-$ by $\frac{|G'|}{|\mathcal{D}_F|}$.
- $d_s^-(G,G') \leq \rho_s$ and $d_t^-(G,G') \leq \rho_t < d_t^+(G,G')$: we insert G' into $G.par$.
- $d_s^-(G,G') \leq \rho_s < d_s^+(G,G')$ and $d_t^-(G,G') \leq \rho_t$: we insert G' into $G.par$.

Regarding $G.div^+$, notice that its value only increases across iterations of the greedy algorithm, as new posts are inserted in the result set $\mathcal{R}$. Therefore, at the end of an iteration, assuming D^* is inserted in $\mathcal{R}$, we update $G.div^+$ as:

$$G.div^+ \leftarrow G.div^+ + w \cdot \frac{\texttt{maxdist}(G.R, D^*.loc)}{\sigma_{max}} + (1-w) \cdot \frac{\texttt{maxdist}(G.T, D^*.t)}{\sigma_{max}}. \tag{8}$$

We are now ready to present the IDX algorithm, whose pseudocode is shown in Algorithm 2. As in BSL, the first step is to retrieve the relevant posts using the methodology from Section 4.1 (line 1). Then these are clustered into the set of groups $\mathcal{G}$ (line 2). The exact partitioning depends on the underlying index structure; we briefly discuss this later. The next step is to compute the coverage information associated with each group (lines 3–7). In particular, for each group G, the spatial and temporal bounds are computed (line 6) and the value $G.cov^-$ and set $G.par$ is updated according to the three cases described earlier.

Subsequently, the main loop of the algorithm begins (lines 10–36), where at the end of each iteration a new post is added to the result set $\mathcal{R}^*$ until k posts are selected. Note that there are three primary data structures in IDX; the set of groups $\mathcal{G}$, the set of seen posts $\mathcal{D}_{seen}$, and the heap H which directs the examination of groups in a best-first manner. Initially, $\mathcal{G}$ contains all groups, and $\mathcal{D}_{seen}$ is empty. In the heap, an entry $\langle g(G)^+, G \rangle$ for each group G is inserted, where the upper bound on the marginal gain $g(G)^+$ is the key, and is computed from Equation 7 (lines 12–14). Also, in the heap, an entry $\langle g(D), D \rangle$ is inserted for each post D having key its current marginal gain $g(D)$.

The inner loop (lines 17–29) examines entries from the heap H until the top entry with the highest key (corresponding to marginal gain or an upper bound thereof) belongs to a post (line 17). At that point, this entry $\langle g(D^*), D^* \rangle$ is deheaped (line 31), and the corresponding post is inserted in the result set (line 32). Because a new result has just been found, the information regarding the diversity of all groups (lines 33–34) and all seen posts, except D^*, (lines 35–37) is updated.

In an iteration of the inner loop (lines 17–30), where entry $\langle g(G)^+, G \rangle$ is deheaped, the following takes place. The group is removed from the set $\mathcal{G}$ of groups (line 19), and all its posts are inserted in set $\mathcal{D}_{seen}$ (line 21). Moreover,

Algorithm 2: Algorithm `IDX`

```
   Input: document collection $\mathcal{D}$, STK filter $F$, result set size $k$
   Output: coverage and diversity aware result set $\mathcal{R}^*$
1  $\mathcal{D}_F \leftarrow$ FindRelevantDocs($\mathcal{D}, F$)                ▷ Section 4.1
2  cluster $\mathcal{D}_F$ into a set of groups $\mathcal{G}$                     ▷ index dependent
3  foreach group $G \in \mathcal{G}$ do
4      $G.cov^- \leftarrow 0$; $G.par \leftarrow \emptyset$
5      foreach group $G' \in \mathcal{G}$ such that $G' \neq G$ do
6          compute bounds $d_s^-(G, G')$, $d_s^+(G, G')$, $d_t^-(G, G')$,
             $d_t^+(G, G')$
7          update $G.cov^-$ and $G.par$ according to the three cases
8  $\mathcal{D}_{seen} \leftarrow \emptyset$                                     ▷ set of seen posts
9  $\mathcal{R}^* \leftarrow \emptyset$
10 while $|\mathcal{R}^*| < k$ do
11     $H \leftarrow \emptyset$                                                  ▷ initialize heap
12     foreach group $G \in \mathcal{G}$ do
13         compute $g(G)^+$                                                       ▷ Equation 7
14         enheap in $H$ entry $\langle g(G)^+, G \rangle$
15     foreach document $D \in \mathcal{D}_{seen}$ do
16         enheap in $H$ entry $\langle g(D), D \rangle$
17     while $H.top$ is a group entry do
18         deheap from $H$ top entry $\langle g(G)^+, G \rangle$
19         $\mathcal{G} \leftarrow \mathcal{G} \setminus \{G\}$
20         foreach document $D \in G$ do
21             $\mathcal{D}_{seen} \leftarrow \mathcal{D}_{seen} \cup \{D\}$
               ▷ compute the coverage of $D$
22             $cov(D) \leftarrow G.cov^-$
23             foreach document $D' \in G' \in G.par$ do
24                 if $d_s(D, D') \leq \rho_s$ and $d_t(D, D') \leq \rho_t$ then
25                     $cov(D) \leftarrow cov(D) + \frac{1}{|\mathcal{D}_F|}$
               ▷ compute the diversity of $D$
26             $div(D) \leftarrow 0$
27             foreach document $D' \in \mathcal{R}^*$ do
28                 $div(D) \leftarrow div(D) + div(D, D')$
               ▷ compute the marginal gain of $D$
29             $g(D) = \frac{1-\lambda}{k} \cdot cov(D) + \frac{\lambda}{k \cdot (k-1)} div(D)$
30             enheap in $H$ entry $\langle g(D), D \rangle$
31     deheap from $H$ top entry $\langle g(D^*), D^* \rangle$
32     $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \{D^*\}$
33     foreach $G \in \mathcal{G}$ do
34         update $G.div^+$                                                       ▷ Equation 8
35     $\mathcal{D}_{seen} \leftarrow \mathcal{D}_{seen} \setminus D^*$
36     foreach $D \in \mathcal{D}_{seen}$ do
37         $g(D) \leftarrow g(D) + \frac{\lambda}{k \cdot (k-1)} div(D, D^*)$     ▷ update $g(D)$
```

for each post $D \in \mathcal{G}$, its exact coverage $cov(D)$ (lines 22–25), its diversity $div(D)$ (lines 26–28), and ultimately its marginal gain $g(D)$ (line 29) are computed. When computing the coverage of D, its group coverage information, $G.cov^-$ and $G.par$, is used to speed up the process. Then an entry $\langle g(D), D \rangle$ for this post is enheaped (line 30).

RCA-based Implementation.

The underlying index structure determines how relevant posts are grouped together. In the inverted index-based RCA approach, posts are spatio-temporally clustered based on their Z-order value. Therefore, a group contains relevant posts that have the same Z-order value.

I^3-based Implementation.

In the I^3 index, posts are grouped together in Octree spatio-temporal cells. Therefore, a group contains relevant posts that reside in the same Octree cell.

5. EXPERIMENTAL EVALUATION

In this section, we present an experimental evaluation of our approach, using two large-scale, real-world datasets of geotagged tweets and photos. We first discuss the experimental setup, outlining the datasets, queries and parameters used in the experiments, and then we present the results.

5.1 Datasets

Next we describe the two datasets used in the experiments. The first dataset is a collection of geotagged tweets that has also been used in [8] and is provided by the authors[1]. It comprises 20M tweets between April and December 2012. The second dataset comprises photos from Flickr, and is provided by Yahoo! [24]. From the original data, we collected a subset of 20M geotagged photos with dates between 2010 and 2014. In both datasets, the posts have a worldwide coverage. The number of distinct keywords is approximately 1.8M for Twitter and 1.3M for Flickr, whereas the average number of keywords per post is 5.7 and 8.4, respectively. The detailed characteristics of the datasets are shown in Table 1. The table also shows the disk space required to store the raw files as well as the constructed indices, both for the I^3-based and the RCA-based implementations. Note that these values refer to the extended versions of those indices that include also the time dimension. Moreover, to evaluate the scalability of our approach, we additionally sampled five subsets from each dataset, with sizes ranging from 4M to 20M.

5.2 Queries and Parameters

To create a set of realistic and meaningful queries for the above datasets, we combined search terms found in trending Twitter topics in 2012[2] as well as popular tags used in Flickr[3]. The goal was to construct queries that reflect exploratory search, having a few hundreds or thousands of results distributed across space and time. Thus, we selected 10 queries, each one having in turn 3 variants, comprising, respectively, 1, 2 or 3 keywords. The queries used are listed in Table 2. In the experiments, we assume OR semantics when using more than one keywords in the query, in order to increase the number of relevant posts. Table 3 lists the average number of relevant posts for these queries in the Twitter and Flickr datasets (for default values of the spatial and temporal filters R and T).

In addition to query keywords, we also vary the size of the spatial and temporal filters. For the former, we use 5 bounding boxes of increasing sizes over the U.S., covering an area ranging, approximately, from 4 million km^2 up to 12 million km^2. For the latter, we use 5 time intervals starting on 01/08/2012 and having duration from 15 up to 75 days. Moreover, we vary the parameter k, i.e. the size of the diversified result subset, from 20 up to 100. Finally, we experimented with different values for the thresholds ρ_s and ρ_t. These settings are summarized in Table 4 (default values are shown in bold).

5.3 Results

Next, we present the results of our experimental evaluation. Specifically, we compare the following four methods: (a) the baseline approach over the I^3-based index (`BSL-I`^{3}) and the RCA-based index (`BSL-RCA`) and (b) our proposed index aware approach over the I^3-based index (`IDX-I`^{3}) and the RCA-based index (`IDX-RCA`). All algorithms were implemented in Java. In particular, for the I^3 and RCA indices we

[1] http://www.ntu.edu.sg/home/gaocong/datacode.htm

[2] https://2012.twitter.com/en/trends.html

[3] https://www.flickr.com/photos/tags/

Table 1: Datasets used in the experiments.

Dataset	Number of geotagged posts	Number of distinct keywords	Average number of keywords	Temporal coverage	Spatial coverage	Disk storage	Index size ($\mathbf{I^3}$-based)	Index size (RCA-based)
Twitter	20M	1,836,679	5.7	Apr.-Dec. 2012	Worldwide	1.5GB	29GB	11GB
Flickr	20M	1,306,785	8.4	2010-2014	Worldwide	2.3GB	79GB	16GB

Table 2: Queries used in the experiments.

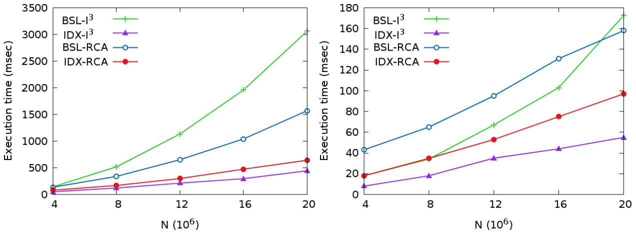
Query	Term 1	Term 2	Term 3
Q_1	obama	election	president
Q_2	olympic	games	london
Q_3	iphone	apple	ipod
Q_4	nascar	race	car
Q_5	kindle	amazon	ebook
Q_6	nba	basketball	sports
Q_7	economy	market	trading
Q_8	war	weapons	violence
Q_9	concert	festival	show
Q_{10}	vacation	summer	trip

Table 3: Average number of relevant posts.

Dataset	$\lvert\Psi\rvert = 1$	$\lvert\Psi\rvert = 2$	$\lvert\Psi\rvert = 3$
Twitter	2,891	6,461	13,395
Flickr	486	1,021	1,699

Table 4: Parameters used in the experiments.

Parameter	Values
Number of geotagged posts (N) (10^6)	4, 8, 12, 16, **20**
Number of query keywords ($\lvert\Psi\rvert$)	1, **2**, 3
Spatial filter size (R) ($10^6 km^2$)	(approx.) 4, 6, **8**, 10, 12
Temporal filter size (T) (days)	15, 30, **45**, 60, 75
Size of diversified result subset (R^k)	20, 40, **60**, 80, 100
Spatial coverage threshold (ρ_s) (%)	2, 4, **6**, 8, 10
Temporal coverage threshold (ρ_t) (%)	2, 4, **6**, 8, 10



(a) Twitter (b) Flickr

Figure 2: Time vs. dataset size.

extended the code that was kindly provided by the authors of [31, 30]. The experiments were conducted on a server with 48GB memory and Intel Xeon E5-2420 v2 processor, running Ubuntu 14.04. In each experiment, we vary one of the parameters listed in Table 4, setting the rest to their default values. The execution time is then measured by executing each of the 10 queries listed in Table 2 5 times and reporting the average.

5.3.1 Increasing the dataset size

First, we evaluate the scalability of our approach by gradually increasing the dataset size. For this purpose, we have sampled both datasets, Twitter and Flickr, creating five subsets for each, with sizes varying from 4M to 20M posts. The results for the average query execution time are shown in Figure 2.

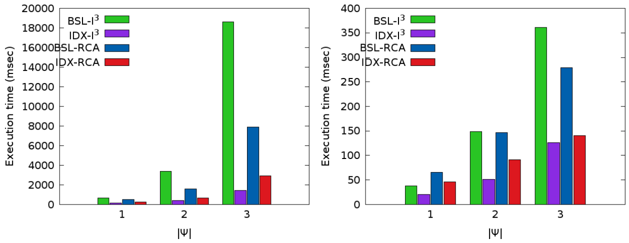
For all methods, execution time increases with the size of the dataset. However, the index aware approach shows much better scalability compared to the baseline. This observation is particularly evident for the Twitter dataset, while less so for the Flickr dataset. The reason for this has to do with the different selectivity of the queries in the two datasets (see also the discussion in Section 5.3.2). Focusing for example on the I^3-based implementation for Twitter, we can observe the following. Although the average query latency for `BSL-I`^{3} starts at below 0.5 seconds, it quickly increases reaching up to more than 3 seconds, whereas at the same time `IDX-I`^{3} still remains below 0.5 seconds. This better scalability of the index aware method results from the fact that it exploits the underlying index structure to effectively prune large portions of the posts that do not contribute to the final result.

Another interesting observation from the Twitter dataset comes from examining the relative performance of the two different implementations. First, comparing the two baselines, we can see that `BSL-RCA` performs better than `BSL-I`^{3}. Since the baseline method does not exploit the underlying index, this can be attributed to the fact that the STK filter is evaluated faster with the RCA-based index. However, for the index aware method, we can see that although both `IDX-I`^{3} and `IDX-RCA` clearly outperform their respective baselines, the difference is even higher for `IDX-I`^{3}, which appears eventually to be slightly faster than `IDX-RCA`. This indicates that the index aware method is able to effectively exploit the underlying index in both cases, but the gain is even higher for the I^3-based index. This can be attributed to the fact that the I^3-based index is more effective during spatio-temporal filtering, whereas the RCA-based index, relying on the Z-order encoding, has the additional overhead of filtering out the false positives. This behavior appears to be consistent also for the rest of the experiments described below.

5.3.2 Decreasing the selectivity of the conditions in the query

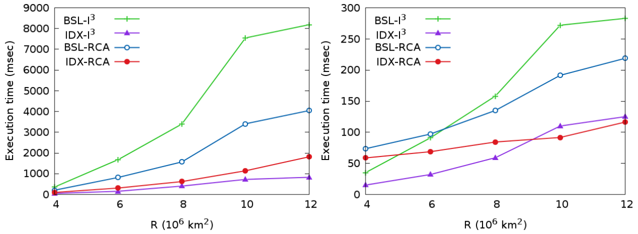
Next, we examine the effect of changing the selectivity of the query filters. This involves three subsets of experiments, corresponding to each of the dimensions addressed: (a) increasing the number of keywords, (b) increasing the size of the spatial region, and (c) increasing the size of the time window. Each of these conditions is examined separately, and the results are shown in Figures 3, 4 and 5, respectively. Note that increasing the number of keywords (under OR semantics), as well as the size of the spatial or the temporal filter, essentially have the same main effect: the number of relevant posts that match with the STK filter of the query increases. In other words, this increases the size of the original result set, from which the top-k results have to be selected.

In all experiments, the index aware methods clearly outperform their respective baselines. More specifically, when the selectivity of the filters is high, the differences are smaller, since the baseline method achieves comparable performance,



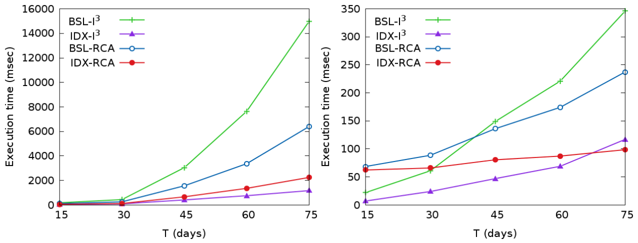
(a) Twitter (b) Flickr

Figure 3: Time vs. number of keywords.



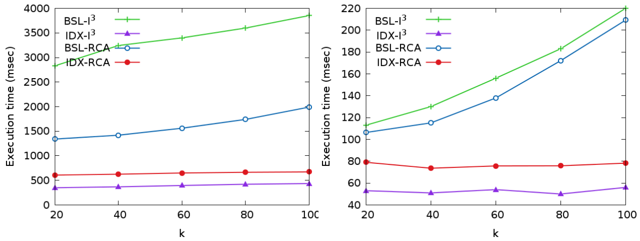
(a) Twitter (b) Flickr

Figure 4: Time vs. spatial region size.



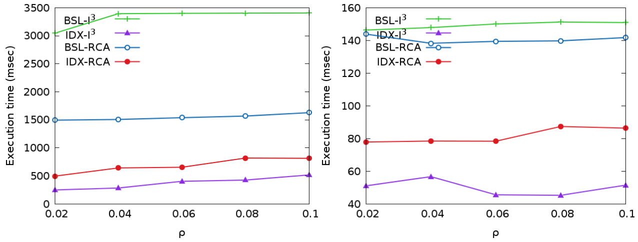
(a) Twitter (b) Flickr

Figure 5: Time vs. time window size.



(a) Twitter (b) Flickr

Figure 6: Time vs. number of results.



(a) Twitter (b) Flickr

Figure 7: Time vs. coverage thresholds.

having to deal with relatively few relevant posts. However, this drastically changes as soon as the filters start to become less selective, allowing for more posts to match. For example, consider the case of the Twitter dataset. Although the average query latency for `BSL-I`^{3} is initially below 1 second, it quickly increases up to 10 seconds or more as the selectivity of the filters decreases. In contrast, `IDX-I`^{3} is significantly less affected, with the average query latency in this case remaining within 1 or 2 seconds, even when the filters reach up to 3 keywords, 12 million km^2 or 75 days. Similar observations can be made also for the Flickr dataset. In that case, although the absolute values of query latency are overall lower, the same differences and trends can be clearly observed. This behavior demonstrates the effectiveness of the pruning strategies and in particular the benefit of using the underlying index structure to prune a large number of comparisons when the size of the original set of relevant posts becomes higher.

5.3.3 *Increasing the number of results*

For the next experiment, we evaluate the effect of the parameter k. The results are shown in Figure 6. For all cases, the index aware methods achieve significant gains over their respective baselines. For instance, for the Twitter dataset, the average query latency for `IDX-I`^{3} and `IDX-RCA` remains below 1 second, while reaching up to 4 seconds for `BSL-I`^{3}. The differences are similar for the Flickr dataset as well, with the baseline methods exhibiting even worse scalability in this case. Interestingly, the performance of the index aware method appears to not be significantly affected by the increase of k. This can be attributed to the fact that, as mentioned in Section 4.2, during the iterations that select the next result to be included in the top-k set, some computed values can be cached and reused in subsequent iterations. Thus, although increasing k means that more iterations have to be performed, the additional cost that is incurred gradually decreases.

Regarding the comparison between the I^3-based and RCA-based implementations, here we can clearly observe a similar behavior as discussed in Section 5.3.1. For the baseline method, the RCA-based index again performs better, requiring less time to apply the STK filter. However, this difference is eventually overcome as the index aware method is again able to more effectively exploit the I^3-based index. Thus, the final result is reversed, with `IDX-I`^{3} achieving the best performance.

5.3.4 *Varying the coverage thresholds*

Finally, we examine the effect of the spatial and temporal thresholds, ρ_s and ρ_t, which determine the radius for coverage for each document. The results are shown in Figure 7. Again, query latency is significantly lower for the index aware methods compared to the baselines. In addition, we can see that the baseline methods do not seem to be affected by this parameter, since the number of comparisons that need to be performed is not affected by these values. Interestingly, this can also be observed for the index aware methods. Notice that these thresholds can be set at query time, thus the underlying index structure is constructed independently of them. Hence, this observation shows that the proposed approach is robust, in the sense that it does not require to fine tune the underlying index according to these thresholds in order to achieve a benefit through the pruning.

Moreover, comparing the performance of the I^3-based and RCA-based implementations, the results are consistent with the findings of the previous experiments. Again, `BSL-RCA` shows an advantage over BSL-I^3, but IDX-I^3 achieves the best performance, having a higher gain that overcomes also this initial difference.

6. CONCLUSIONS

In this paper, we have introduced a novel type of spatial-temporal-keyword query, the kCD-STK query. This query is based on two key notions, *spatio-temporal coverage* and *diversity*, which are formally defined. In particular, the query is formulated similarly to other search results diversification problems, which allows us to derive a baseline approach for its evaluation. Then, we focus on developing a more efficient strategy for processing kCD-STK qeuries, which allows to exploit an underlying hybrid (spatial-temporal-keyword) index not only for the first part, i.e. the filtering of relevant posts, but also for the second part, i.e. the coverage and diversity aware selection of the top-k results. To that end, we have considered two state-of-the-art spatio-textual indices, which we extended to include also the time dimension, and we have shown how our proposed index aware approach can be applied on top of those structures.

To validate and evaluate our approach, we have conducted an experimental evaluation on large real-world datasets containing geotagged tweets and photos. The results have shown that our optimized approach manages to successfully exploit the available index to significantly reduce the query execution time compared to the baseline algorithm. This holds for both indices that have been considered, namely the I^3-based and the RCA-based implementations.

In the future, an interesting direction is to investigate how the proposed concepts and approach can be applied when dealing with streams of spatio-temporal posts.

Acknowledgements

This work was partially supported by the EU Project City.Risks (H2020-FCT-2014-653747).

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *WSDM*, pages 5–14, 2009.

[2] O. Alonso and K. Shiells. Timelines as summaries of popular scheduled events. In *WWW '13 Companion*, pages 1037–1044, 2013.

[3] O. Alonso, J. Strötgen, R. A. Baeza-Yates, and M. Gertz. Temporal information retrieval: Challenges and opportunities. In *WWW Workshop on Linked Data on the Web*, pages 1–8, 2011.

[4] A. Borodin, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *PODS*, pages 155–166, 2012.

[5] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. Survey of temporal information retrieval and related applications. *ACM Comput. Surv.*, 47(2):15:1–15:41, 2014.

[6] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[7] L. Chen and G. Cong. Diversity-aware top-k publish/subscribe for text stream. In *SIGMOD*, pages 347–362, 2015.

[8] L. Chen, G. Cong, X. Cao, and K. Tan. Temporal spatial-keyword top-k publish/subscribe. In *ICDE*, pages 255–266, 2015.

[9] L. Chen, G. Cong, C. S. Jensen, and D. Wu. Spatial keyword query processing: An experimental evaluation. *PVLDB*, 6(3):217–228, 2013.

[10] Y. Chen, H. Amiri, Z. Li, and T. Chua. Emerging topic detection for organizations from microblogs. In *SIGIR*, pages 43–52, 2013.

[11] Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *SIGMOD*, pages 277–288, 2006.

[12] M. Christoforaki, J. He, C. Dimopoulos, A. Markowetz, and T. Suel. Text vs. space: efficient geo-search query processing. In *CIKM*, pages 423–432, 2011.

[13] Z. Dou, S. Hu, K. Chen, R. Song, and J.-R. Wen. Multi-dimensional search result diversification. In *WSDM*, pages 475–484, 2011.

[14] M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, 39(1):41–47, 2010.

[15] M. Drosou and E. Pitoura. Multiple radii DisC diversity: Result diversification based on dissimilarity and coverage. *ACM Trans. Database Syst.*, 40(1):4, 2015.

[16] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *J. Comput. Syst. Sci.*, 66(4):614–656, 2003.

[17] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., 1990.

[18] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW*, pages 381–390, 2009.

[19] A. Jatowt, É. Antoine, Y. Kawai, and T. Akiyama. Mapping temporal horizons: Analysis of collective future and past related attention in twitter. In *WWW*, pages 484–494, 2015.

[20] N. Kanhabua and W. Nejdl. Understanding the diversity of tweets in the time of outbreaks. In *WWW*, pages 1335–1342, 2013.

[21] S. Nepomnyachiy, B. Gelley, W. Jiang, and T. Minkus. What, where, and when: keyword search with spatio-temporal ranges. In *GIR*, pages 2:1–2:8, 2014.

[22] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

[23] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *SIGKDD*, pages 1104–1112, 2012.

[24] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.

[25] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson. Spatio-textual indexing for geographical search on the web. In *SSTD*, pages 218–235, 2005.

[26] G. Valkanas and D. Gunopulos. How the live web feels about events. In *CIKM*, pages 639–648, 2013.

[27] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia. Efficient computation of diverse query results. In *ICDE*, pages 228–236, 2008.

[28] M. R. Vieira, H. L. Razente, M. C. N. Barioni, M. Hadjieleftheriou, D. Srivastava, C. T. Jr., and V. J. Tsotras. On query result diversification. In *ICDE*, pages 1163–1174, 2011.

[29] D. Wu, G. Cong, and C. S. Jensen. A framework for efficient spatial web object retrieval. *VLDB J.*, 21(6):797–822, 2012.

[30] D. Zhang, C. Chan, and K. Tan. Processing spatial keyword query as a top-k aggregation query. In *SIGIR*, pages 355–364, 2014.

[31] D. Zhang, K.-L. Tan, and A. K. Tung. Scalable top-k spatial keyword search. In *EDBT*, pages 359–370, 2013.

[32] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W. Ma. Hybrid index structures for location-based web search. In *CIKM*, pages 155–162, 2005.