PAUSANIAS: Final activity report

Akrivi Vlachou, Postdoctoral Researcher

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

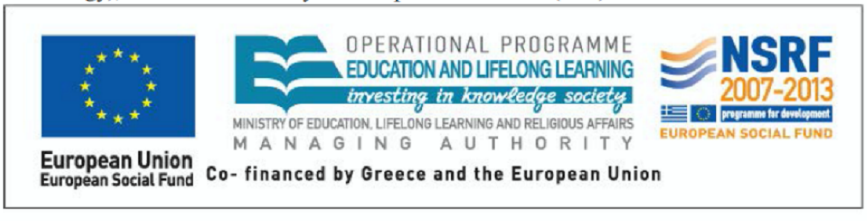
Abstract

Search engines, such as Google and Yahoo!, provide efficient retrieval and ranking of web pages based on queries consisting of a set of given keywords. Recent studies show that 20% of all Web queries also have location constraints, i.e., also refer to the location of a geotagged web page. An increasing number of applications support location-based keyword search, including Google Maps, Bing Maps, Yahoo! Local, and Yelp. Such applications depict points of interest on the map and combine their location with the keywords provided by the associated document(s). The posed queries consist of two conditions: a set of keywords and a spatial location. The goal is to find points of interest with these keywords close to the location. We refer to such a query as spatial-keyword query. Moreover, mobile devices nowadays are enhanced with built-in GPS receivers, which permits applications (such as search engines or yellow page services) to acquire the location of the user implicitly, and provide location-based services. For instance, Google Mobile App provides a simple search service for smartphones where the location of the user is automatically captured and employed to retrieve results relevant to her current location. As an example, a search for pizza results in a list of pizza restaurants nearby the user. In this research project, we studied how preference queries can be extended for supporting also keywords.

To this end we first studied preference queries in order to establish techniques that can be extended for supporting keywords (Chapter 1). Moreover, we proposed Top-k Spatio-Textual Preference Queries and proposed a novel indexing scheme and two algorithms for supporting efficient query processing (Chapter 2). We also studied the problem of maximizing the influence of spatio-textual objects based on reverse top-k queries and keyword selection (Chapter 3). Finally, we analyze the properties of geotagged photos of Flickr, and propose novel location-aware tag recommendation methods (Chapter 4). In summary, this research project lead to the following publications:

- George Tsatsanifos, Akrivi Vlachou: On Processing Top-k Spatio-Textual Preference Queries, in Proceedings of 18th International Conference on Extending Database Technology (EDBT), Brussels, Belgium, March 23-27, 2015.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



- Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis and Kjetil Nørvåg: Finding the Most Diverse Products using Preference Queries, in Proceedings of 18th International Conference on Extending Database Technology (EDBT), Brussels, Belgium, March 23-27, 2015.
- Ioanna Miliou, Akrivi Vlachou: Location-Aware Tag Recommendations for Flickr, DEXA (1) 2014: 97-104.
- Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis and Kjetil Nørvåg: Efficient Processing of Exploratory Top-k Joins, in Proceedings of 26th International Conference on Scientific and Statistical Database Management (SSDBM), Aalborg, Denmark, June 30 - July 2, 2014.
- Akrivi Vlachou, Christos Doulkeridis, Kjetil Nørvåg and Yannis Kotidis: Branch-and-Bound Algorithm for Reverse Top-k Queries, in Proceedings of ACM International Conference on Management of Data (SIGMOD), New York, USA, June 22-27, 2013.
- Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis and Kjetil Nørvåg: Discovering Influential Data Objects over Time, in Proceedings of 13th International Symposium on Spatial and Temporal Databases (SSTD), Munich, Germany, August 21-23, 2013.
- Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis, Kjetil Nørvåg: Maximizing Influence of Spatio-Textual Objects through Keyword Selection submitted for publication.
- Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis, Kjetil Nørvåg: Exploratory product search using top-k join queries. submitted for journal publication.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Contents

1 Preference Queries

1.1 Branch-and-Bound Algorithm for Reverse Top-k Queries . . . . .
1.2 Discovering Influential Data Objects over Time . . . . . .
1.3 Efficient Processing of Exploratory Top-k Joins . . . . . .
1.4 Finding the Most Diverse Products using Preference Queries . . .

2 On Processing Top-k Spatio-Textual Preference Queries

2.1 Introduction . . . . . .
2.2 Related Work . . . . . .
2.3 Problem Statement . . . . . .
2.4 Indexing . . . . . .
2.4.1 Index Characteristics . . . . . .
2.4.2 Indexing based on Hilbert Mapping . . . . . .
2.5 Spatio-Textual Data Scan (STDS) . . . . . .
2.6 Spatio-Textual Preference Search (STPS) . . . . . .
2.6.1 Valid Combination of Feature Objects . . . . . .
2.6.2 STPS Overview . . . . . .
2.6.3 Spatio-Textual Feature Objects Retrieval . . . . . .
2.6.4 Retrieval of Qualified Data Objects . . . . . .
2.7 Variants of Top-k Spatio-Textual Preference Queries . . . . . .
2.7.1 Influence-Based STPQ Queries . . . . . .
2.7.2 Nearest Neighbor STPQ Queries . . . . . .
2.8 Experimental Evaluation . . . . . .
2.8.1 Experimental Setup . . . . . .
2.8.2 Scalability Analysis . . . . . .
2.8.3 Varying Query Parameters . . . . . .
2.8.4 Influence-based Preference Score . . . . . .
2.8.5 Nearest Neighbor Preference Score . . . . . .

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

Co- financed by Greece and the European Union

2.9 Conclusions . . . .
2.9.1 Acknowledgments . . . .

3 Maximizing Influence Of Spatio-Textual Objects Through Keyword Selection
3.1 Introduction . . . .
3.2 Related Work . . . .
3.3 Preliminaries . . . .
3.3.1 Top-k spatial keyword queries . . . .
3.3.2 IR-tree . . . .
3.4 Problem Definition . . . .
3.5 Baseline . . . .
3.6 Graph Based Term Selection . . . .
3.7 Experimental Evaluation . . . .
3.8 Conclusions . . . .
3.8.1 Acknowledgments . . . .

4 Location-aware Tag Recommendations for Flickr
4.1 Introduction . . . .
4.2 Data Collection . . . .
4.2.1 Distribution of Tag Frequency . . . .
4.2.2 Distribution of Number of Tags per Photo . . . .
4.2.3 Analysis based on WordNet . . . .
4.3 Recommendation Methods . . . .
4.3.1 System Overview . . . .
4.3.2 Tag Recommendation Methods . . . .
4.4 Experimental Evaluation . . . .
4.4.1 Prototype System . . . .
4.4.2 Experimental Evaluation . . . .
4.5 Related Work . . . .
4.6 Conclusions . . . .
4.6.1 Acknowledgments . . . .

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Chapter 1

Preference Queries

Initially, we identified several interesting problems that are not addressed yet and are important for discovering interesting data objects. This relates to spatial keyword search, because the user wants to retrieve relevant data not only based on the textual data, but also on the spatial data. The restriction on the expressiveness and the lack of efficient algorithms of ranked queries limits also the expressiveness and the efficiency of spatial-keyword search. Thus, we first focus on improving reverse top-k queries and extending top-k queries. Top-k queries return to the user only the k best objects based on the individual user preferences and comprise an essential tool for rank-aware query processing. Assuming a stored data set of user preferences, reverse top-k queries have been introduced for retrieving the users that deem a given database object as one of their top-k results. Reverse top-k queries have already attracted significant interest in research, due to numerous real-life applications such as market analysis and product placement. However, the best existing algorithm for computing the reverse top-k query is not efficient for all data distributions. Thus, we developed novel algorithms for efficient processing of reverse top-k queries that can be easily adapted in the context of spatial-keyword search (Section 1.1). Moreover, an important characteristic of spatial-keyword search is that it involves data that are available through the World Wide Web and that continuously change over time. Thus, it is very important to take into account the temporal dimension of the data. For this purpose, we define the continuous influential query, which retrieves the object that remains influential for the longest temporal range within a time horizon based on the reverse top-k queries (Section 1.2). Then, we address the problem of discovering a ranked set of k distinct main objects combined with additional (accessory) objects that best fit the given preferences. We model this problem as a rank-join problem where

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

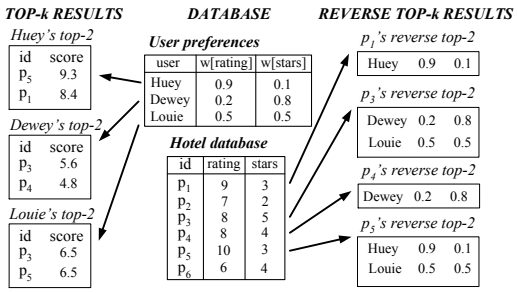


Figure 1.1: Example of reverse top-k queries.

each combination is represented by a set of tuples from different relations and we call the respective query *eXploratory Top-k Join* query (Section 1.3). Finally, we address the problem of discovering a bounded set of r diverse products that attract the interests of different customers. This problem finds numerous applications in electronic marketplaces, e.g., for selecting the products that are placed in the home page of an online shop (Section 1.4), which in turn makes it very interesting for the case where also keywords exist.

1.1 Branch-and-Bound Algorithm for Reverse Top-k Queries

Given a database of objects described by a set of numerical scoring attributes and a user with a preference function defined over these attributes, a top-k query retrieves the k objects with best score for the particular preference function. In the model that is widely used in related work [10, 22] and in practice, the users express their preferences through linear top-k queries, which are defined by assigning a weight to each of the scoring attributes, indicating the importance of each attribute to the user. Assuming a stored data set of user preferences, reverse top-k queries have been proposed [46, 47] to retrieve the user preferences that make a given object belong to the respective top-k result set. From the perspective of a manufacturer, it is important to identify the customers who are potentially interested in her products and to estimate the visibility of a product based on the different user preferences for which it appears in the top-ranked positions. Hence, reverse top-k queries comprise an essential tool for business analysis, allowing manufacturers to assess the impact of their products in the market based on the competition.

More formally, a reverse top-k query returns for a point q and a positive in-

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

teger k, the set of linear preference functions (in terms of weighting vectors) for which q is contained in their top-k result. Consider for example a database containing information about different hotels as well as user preferences, as depicted in Figure 2.1. For each of the six hotels, the rating and the number of stars are recorded, and maximum values on each attribute are preferable. The database also stores the preferences of three users (Huey, Dewey and Louie) in terms of weights on each attribute. Different users may have different preferences about a potential hotel. For instance, Huey prefers hotels with high rating values, whereas Dewey is interested in hotels with many stars. Louie is indifferent or values equally rating and stars. On the left part of the figure, the top-2 hotels are depicted for each user along with their scores. On the right part, the reverse top-2 results are shown for the hotels. Notice that p_2 and p_6 have empty reverse top-2 result sets, i.e., they do not belong to the top-2 list of any user.

Currently, the most efficient algorithm for computing the reverse top-k set is the RTA algorithm [46]. RTA has two main drawbacks when processing a reverse top-k query: (i) it needs to access all stored user preferences, and (ii) it cannot avoid executing a top-k query for each user preference (determined by the corresponding user weights) that belongs to the result set. As a result, the performance of RTA is sensitive to the cardinality of the reverse top-k result; for queries with result sets of high cardinality RTA often becomes inefficient. Since we expect that reverse top-k queries will be posed for query points that are highly ranked, and therefore have a result set of high cardinality, this drawback severely limits the practicality of RTA.

To alleviate the shortcomings of RTA, we study the conditions in which a set of weighting vectors (representing linear preference functions) can be immediately added to the result set. Therefore, we focus on whether a data point may be ranked higher than the query point for a set of weighting vectors. In addition, we address the question whether a set of weighting vectors can be excluded from the reverse top-k result. Based on these properties, we develop an efficient branch-and-bound algorithm assuming that both data sets are indexed by multidimensional access methods.

The contributions of this work are summarized here:

- We introduce useful properties for processing reverse top-k queries without accessing each user's individual preferences nor executing the respective top-k query.
- We present a novel algorithm that processes sets of weighting vectors, without having to examine each vector individually, and use this algorithm as

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

basic building block for our reverse top-k algorithms.

- We propose a framework for reverse top-k query processing that employs the branch-and-bound methodology and exploits the introduced properties.
- We present two optimizations of the basic branch-and-bound algorithm (BBR) that use result sharing ($BBR*$) and an aggregate R-tree ($BBRA$) to boost its performance.
- We conduct a thorough experimental evaluation that demonstrates the efficiency of our proposed algorithms.

For more details refer to: Akrivi Vlachou, Christos Doulkeridis, Kjetil Nørvåg and Yannis Kotidis: Branch-and-Bound Algorithm for Reverse Top-k Queries, in Proceedings of ACM International Conference on Management of Data (SIGMOD), New York, USA, June 22-27, 2013.

1.2 Discovering Influential Data Objects over Time

In online marketplaces, top-k queries are typically used to present a limited number of products ranked according to the user's preferences. This is extremely helpful for the user as it enables decision-making, without the need to inspect large amounts of possibly uninteresting results. In addition, the user is not overwhelmed by the available information and can retrieve results that satisfy her information need. As a result, an increasing amount of research has focused on efficient techniques for top-k query processing lately [24].

From the perspective of the product manufacturers top-k queries are of great interest as well, since the visibility of a product clearly depends on the number of different top-k queries for which it belongs to the result set. The reason for this is twofold: 1) users usually consider only a few highly ranked products and ignore the remaining ones, and 2) products that appear in the top-k result sets are far more likely to be chosen by a potential customer, because these products satisfy the customers' preferences. Recently, *reverse top-k queries* [46] were proposed to study the visibility of a given product. A reverse top-k query returns the set of user preferences (i.e., customers) for which a given product is in the result set of the respective top-k queries. Intuitively, a product that appears in as many as possible top-k result sets, has a higher visibility and therefore also a higher impact on the market. This has naturally lead to the definition of the *most influential products*

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

based on the cardinality of their reverse top-k result sets [48]. Identifying the most influential products from a given set of products is important for market analysis, since the product manufacturer can estimate the impact of her products in the market.

However, an important aspect of a product's influence that has not been taken into account yet is its variance over time as the user preferences change. The customers' criteria can differ significantly over time for various reasons. For example, in online marketplaces, new customers pose queries and new preferences are collected. In addition, customers that have already posed queries will disconnect after some time. As user preferences change over time, a product which appears consistently in the top-k results of as many customers as possible, thus satisfying many customers' criteria at any time, has a higher impact on the market than a product that is absent from those results. Therefore, these products are the best candidate products to advertise to potential customers, and it is important to identify such products efficiently.

In this work, we study for the first time the problem of finding the product that belongs consistently to the most influential products over time, the *continuous influential products*. This is an important problem for many real-life applications. For example, the products advertised on the first page of an online marketplace should be the products that have the greatest impact on the market, i.e., the products that are the most popular among the customers. Since customers change all the time, the products that consistently belong to the most influential products over time are more probable to attract many potential customers at any time. It is therefore essential to identify the objects (products) that have high impact over a period of time and despite the fluctuation of preferences these objects remain among the most influential objects. From now on we will use the terms *product* and *object* interchangeably.

In the following, we first define formally the problem of continuous influential products and provide a baseline algorithm that sequentially scans all time intervals in order to retrieve the most continuous influential product. Then, we provide a bounding scheme in order to facilitate early termination of our algorithms and avoid processing time intervals that do not alter the result set. Summarizing, the main contributions of this work are:

- We study, for the first time, the problem of identifying the data object that has the highest impact over time.
- An appropriate score of influence (called *continuity score*) based on the reverse top-k query is defined to capture the product impact over a period

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

of time.

- We derive upper and lower bounds for the continuity score of a given object that lead to efficient algorithms for retrieving the most continuous influential product. Two different algorithms are presented that provide early termination based on the bounds, but follow different strategies in order to terminate as soon as possible.

- We conduct a detailed experimental study for various setups and demonstrate the efficiency of our algorithms.

For more details refer to: Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis and Kjetil Nørvåg: Discovering Influential Data Objects over Time, in Proceedings of 13th International Symposium on Spatial and Temporal Databases (SSTD), Munich, Germany, August 21-23, 2013.

1.3 Efficient Processing of Exploratory Top-k Joins

Top-k queries [24] are often used to help users select the k best objects according to their preferences from a large set of objects. A product is typically represented by a d-dimensional point p where each dimension describes a specific feature. Usually, preferences are expressed through a weighting vector $\mathbf{w}$ of d dimensions, each corresponding to an attribute of the product, while the value of the dimension indicates the importance of the specific attribute to the user. The ranking of the objects is based on a scoring function $f_{\mathbf{w}}(p)$, and one of the most common ones is the weighted sum $f_w(p) = \sum_i^d \mathbf{w}[i]p[i]$.

To this end, we propose the eXploratory Top-k Join (XTJ_k) query. An XTJ_k takes as input a a set of relations where there is a *main* relation and the rest *additional* relations are joined to the main relation forming a "star"-like structure. Among all possible combinations, only the best for each product are considered and the top-k of them are returned to the user.

Current state-of-the-art techniques [18, 23, 40] for computing combinations based on preference vectors fall short to address this kind of queries as they assume that each result should contain objects from all relations participating in the join. On the contrary, our requirement is that an object should be added to a combination only if it is beneficial for the combination. Moreover, current techniques do not exploit the form of the result-set and the structure of the join, fact that leads to suboptimal performance.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

To summarize, the contributions of this work are: a) we introduce the eXploratory Top-k Join (XTJ_k) query, a novel query type which creates combinations of variable size between main and additional objects and returns the top-k combinations with discrete main objects, b) we introduce an efficient bounding scheme for our algorithm, and c) we perform an experimental evaluation that demonstrates the efficiency of our approach.

For more details refer to: Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis and Kjetil Nørvåg: Efficient Processing of Exploratory Top-k Joins, in Proceedings of 26th International Conference on Scientific and Statistical Database Management (SSDBM), Aalborg, Denmark, June 30 - July 2, 2014.

1.4 Finding the Most Diverse Products using Preference Queries

Top-k queries [44] help customers select a ranked set of k products that best match their preferences out of an overwhelmingly large collection of products. For a specific customer, her preferences are expressed by means of a top-k query, and highly ranked products in the top-k result are more attractive to the customer. Thus, from the perspective of product sellers, the visibility and the potential market of a product relates to the top-k queries for which the product is highly ranked. Towards this direction, reverse top-k queries [46] retrieve the set of user preferences for which a product appears in their top-k lists. Reverse top-k queries are very important for estimating the impact of the product on the market, as the cardinality of the result set defines an *influence score* [49] for the product, i.e., the number of customers that value a particular product.

We study the problem of finding the r most diverse products based on the user preferences. The goal is to find a set of products that are attractive to a wide range of customers with different preferences. For instance, consider an electronic marketplace that wishes to advertise r products on its front page aiming to attract as many new customers as possible. Advertising diverse products that are attractive to different existing customers increases the probability that a new customer finds one of those products attractive. The strategy of advertising the r most influential products [49], i.e., the r products that attract the highest total number of customers, does not necessarily lead to a set of diverse products and may fail to attract many new customers, since such products may be attractive to customers with similar preferences.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

User Preferences:

User	$w[1]$	$w[2]$	$w[3]$	**Top-k**
Bob	0.1	0.2	0.7	p_1
Tom	0.1	0.3	0.6	p_1
Jack	0.3	0.1	0.6	p_2
Max	0.8	0.1	0.1	p_3

Products:

Product	$p[1]$	$p[2]$	$p[3]$	**Reverse top-k**
p_1	1	2	6	Bob,Tom
p_2	2	1	6	Jack
p_3	6	5	2	Max

Table 1.1: Example of product database and user preferences.

Consider for example the set of user preferences and products depicted in Table 1.1, where maximum values in product attributes are preferable. Assume that the goal is to advertise two products for attracting new customers. Our proposed method selects the $r = 2$ most diverse products based on user preferences, which in our example is the set $\{p_1, p_3\}$. This set is more probable to attract more new customers because p_1 and p_3 satisfy more diverse preferences. For example, a customer with similar preferences to Jack is highly probable to be attracted also to p_1, even though it is not the best option for her on the market. This is because both p_1 and p_2 satisfy users that have high preference for the third dimension (expressed with a high weight $w[3]$). On the other hand, p_3 satisfies users that have totally diverse preferences compared to p_1 and p_2, namely users such as Max that prefer the first dimension.

We introduce the problem of finding the r *most diverse* products based on user preferences. The user preferences are captured by the reverse top-k set of each product. We model this problem as a *dispersion* problem [37] using as distance function the dissimilarity of the reverse top-k sets. In this sense, the set of r objects with the maximum diversity is returned to the user. Consequently, the selected objects are appealing to many different customers with dissimilar user preferences. Different from our work, existing solutions for identifying diverse objects rely solely on product attributes and largely overlook user preferences [45]. On the other hand, approaches that identify r objects with high total number of customers [30, 49], often fail to discover truly diverse products that can be appealing

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

to new customers with different preferences than those of the existing ones.

To summarize the contributions of this work are:

- We study the novel problem of finding the r most diverse products based on user preferences. We model this problem as a *dispersion* problem and define an appropriate distance function that captures the dissimilarity of products based on their reverse top-k sets.
- As dispersion problems are known to be NP-hard [16], we use a greedy algorithm that retrieves r diverse products, after computing the reverse top-k sets of the products efficiently.
- To improve the performance of our algorithm, we propose an alternative algorithm that progressively computes an approximation of the reverse top-k sets of a limited set of candidate products and retrieves a set of r products of high diversity.
- We present maintenance techniques for updating the r most diverse products in the case of dynamic data in a cost-efficient way. In addition, we generalize our approach to support any set-based similarity function.
- We demonstrate the efficiency and achieved diversity of our algorithms using both synthetic and real-life data sets.

For more details refer to: Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis and Kjetil Nørvåg: Finding the Most Diverse Products using Preference Queries, in Proceedings of 18th International Conference on Extending Database Technology (EDBT), Brussels, Belgium, March 23-27, 2015.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Chapter 2

On Processing Top-k Spatio-Textual Preference Queries

In this work we propose a novel query type, termed *top-k spatio-textual preference query*, that retrieves a set of spatio-textual objects ranked by the goodness of the facilities in their neighborhood. Consider for example, a tourist that looks for "hotels that have nearby a highly rated Italian restaurant that serves pizza". The proposed query type takes into account not only the spatial location and textual description of spatio-textual objects (such as hotels and restaurants), but also additional information such as ratings that describe their quality. Moreover, spatio-textual objects (i.e., hotels) are ranked based on the features of facilities (i.e., restaurants) in their neighborhood. Computing the score of each data object based on the facilities in its neighborhood is costly. To address this limitation, we propose an appropriate indexing technique and develop an efficient algorithm for processing our novel query. Moreover, we extend our algorithm for processing spatio-textual preference queries based on alternative score definitions under a unified framework. Last but not least, we conduct extensive experiments for evaluating the performance of our methods.

2.1 Introduction

An increasing number of applications support location-based queries, which retrieve the most interesting spatial objects based on their geographic location. Recently, spatio-textual queries have lavished much attention, as such queries combine location-based retrieval with textual information that describes the spatial

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

objects. Most of the existing queries only focus on retrieving objects that satisfy a spatial constraint ranked by their spatio-textual similarity to the query point. However, in addition users are quite often interested in spatial objects (*data objects*) based on the quality of other facilities (*feature objects*) that are located in their vicinity. Feature objects are typically described by *non-spatial* attributes such as quality or rating, in addition to the *textual description*. We propose a novel and more expressive query type than existing spatio-textual queries, called *top-k spatio-textual preference query*, for ranked retrieval of data objects based the textual relevance and the non-spatial score of feature objects in their neighborhood.

Consider for example, a tourist that looks for *"hotels that have nearby a highly rated **Italian** restaurant that serves **pizza**"*. Figure 2.1 depicts a spatial area containing hotels (data objects) and restaurants (feature objects). The quality of the restaurants based on existing reviews is depicted next to the restaurant. Each restaurant also has textual information in the form of keywords extracted from its menu, such as pizza or steak, which describes additional characteristics of the restaurant. The tourist also specifies a spatial constraint (in the figure depicted as a range around each hotel) to restrict the distance of the restaurant to the hotel. Obviously, the hotel h_2 is the best option for a tourist that poses the aforementioned query. In the general case, more than one type of feature objects may exist in order to support queries such as *"hotels that have nearby a good **Italian** restaurant that serves **pizza** and a cheap coffeehouse that serves **muffins**"*. Even though spatial preference queries have been studied before [53, 54, 39], their definition ignores the available textual information. In our example, the spatial preference query would correspond to a tourist that searches for *"hotels that are nearby a good restaurant"* and the hotel h_1 would always be retrieved, irrespective of the textual information.

We define top-k spatio-textual preference queries and provide efficient algorithms for processing this novel query type. A main challenge compared to traditional spatial preference queries [53, 54, 39] is that the score of a data object changes depending on the query keywords, which renders techniques that rely on materialization (such as [39]) not applicable. Most importantly, processing spatial preference queries is costly in terms of both I/O and execution time [53, 54]. Thus, extending spatial preference queries for supporting also textual information is challenging, since the new query type is more demanding due to the additional textual descriptions.

A straightforward algorithm for processing spatio-textual preference queries is to compute the *spatio-textual preference score* for each data object and then report the k data objects with the highest score. We call this approach *Spatio-*

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

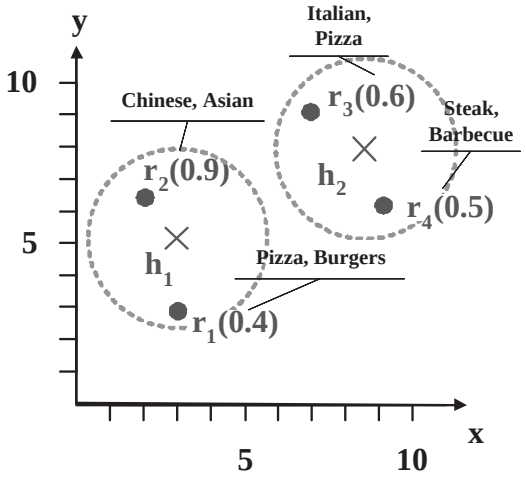


Figure 2.1: Motivating example.

Textual Data Scan (STDS) and examine it as a baseline, while our main focus is to reduce the cost required for computing the spatio-textual score of a data object.

Moreover, we develop an efficient and scalable algorithm, called *Spatio-Textual Preference Search (STPS)*, for processing spatio-textual preference queries. *STPS* follows a different strategy than *STDS*, as it retrieves highly ranked feature objects first, and then searches data objects in their spatial neighborhood. Intuitively, data objects located in the neighborhood of highly ranked feature objects are good candidates for inclusion in the top-k result set. The main challenge tackled with *STPS* is determining efficiently the best feature objects from all feature sets that do not violate the spatial constraint.

To further improve the performance of our algorithms, we develop an appropriate indexing technique called *SRT-index*, that not only indexes the spatial location, the textual description and the non-spatial score, but in addition takes them equally into consideration during the index creation. Finally, we extend our algorithm for processing spatio-textual preference queries based on alternative score definitions under a unified framework. To summarize the contribution of this work are:

- We propose a novel query type, called top-k spatio-textual preference query, that ranks the data objects based on the quality and textual relevance of facilities (*feature objects*) located in their vicinity.
- A novel indexing technique called *SRT-index* is presented that is beneficial

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

for processing spatio-textual preference queries.

- We present two algorithms for processing spatio-textual preference queries, namely *Spatio-Textual Data Scan (STDS)* and *Spatio-Textual Preference Search (STPS)*.
- We extend our algorithm *STPS* for processing spatio-textual preference queries based on alternative score definitions under a unified framework.
- We conduct an extensive experiment evaluation for studying the performance of our proposed algorithms and indexing technique.

The rest of this chapter is organized as follows: Section 4.5 overviews the relevant literature. In Section 2.3, we define the spatio-textual preference query. Our novel indexing technique (*SRT-index*) is presented in Section 2.4. In Section 2.5 we describe our baseline algorithm, called spatio-textual data scan (*STDS*). An efficient algorithm, called Spatio-Textual Preference Search (*STPS*), is proposed in Section 2.6. Moreover, we extend our algorithms for processing spatio-textual preference queries based on alternative scores in Section 2.7. We present the experimental evaluation in Section 2.8 and we conclude in Section 3.8.

2.2 Related Work

Recently several approaches have been proposed for spatial-keyword search. In [17], the problem of distance-first top-k spatial keyword search is studied. To this end, the authors propose an indexing structure (IR^2-Tree) that is a combination of an R-Tree and signature files. The IR-Tree was proposed in another conspicuous work [14, 29], which is a spatio-textual indexing approach that employs a hybrid index that augments the nodes of an R-Tree with inverted indices. The inverted index at each node refers to a pseudo-document that represents all the objects under the node. During query processing, the index is exploited to retrieve the top-k data objects, defined as the k objects that have the highest spatio-textual similarity to a given data location and a set of keywords. Moreover, in [38] the *Spatial Inverted Index (S2I)* was proposed for processing top-k spatial keyword queries. The S2I index maps each keyword to a distinct aggregated R-Tree or to a block file that stores the objects with the given term. All these approaches focus on ranking the data objects based on their spatio-textual similarity to a query point and some keywords. This is different from our work, which ranks the data objects based

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

on textual relevance and a non-spatial score (quality) of the facilities in their spatial neighborhood. [11] provides an all-around evaluation of spatio-textual indices and reports on the findings obtained when applying a benchmark to the indices.

Spatio-textual similarity joins were studied in [4]. Given two data sets, the query retrieves all pairs of objects that have spatial distance smaller than a given value and at the same time a textual similarity that is larger than a given value. This differs from the top-k spatio-textual preferences query, because the spatio-textual similarity join does not rank the data objects and some data objects may appear more than once in the result set. Prestige-based spatio-textual retrieval was studied in [7]. The proposed query takes into account both location proximity and prestige-based text relevance.

The m-closest keywords query [55] aims to find the spatially closest data objects that match with the query keywords. The authors in [8] study the spatial group keyword query that retrieves a group of data objects such that all query keywords appear in at least one data object textual description and such that objects are nearest to the query location and have the lowest inter-object distances. These approaches focus on finding a set of data objects that are close to each other and relevant to a given query, whereas in this work we rank the data objects based on the facilities in their spatial neighborhood. In [9], the length-constrained maximum-sum region (LCMSR) query is proposed that returns a spatial-network region of constrained size that is located within a general region of interest and that best matches query keywords.

Ranking of data objects based on their spatial neighborhood without supporting keywords has been studied in [52, 15, 53, 54, 39]. Xia *et al.* studied the problem of retrieving the top-k most influential spatial objects [52], where the score of a data object p is defined as the sum of the scores of all feature objects that have p as their nearest neighbor. Yang *et al.* studied the problem of finding an optimal location [15], which does not use candidate data objects but instead searches the space. Yiu *et al.* first considered computing the score of a data object p based on feature objects in its spatial neighborhood from multiple feature sets [53, 54] and defined top-k spatial preference queries. In another line of work, a materialization technique for top-k spatial preference queries was proposed in [39] which leads to significant savings in both computational and I/O cost during query processing. The main difference is that our novel query is defined in addition by a set of keywords that express desirable characteristics of the feature objects (like “pizza” for a feature object that represents a restaurant).

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program ”Education and Lifelong Learning” (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

2.3 Problem Statement

Given an *object* dataset O and a set of c *feature* datasets $\{F_i \mid i \in [1, c]\}$, we address the problem of finding k data objects that have in their spatial proximity highly ranked feature objects that are relevant to the given query keywords. Each data object $p \in O$ has a spatial location. Similarly, each feature object $t \in F_i$ is associated with a spatial location but also with a *non-spatial score* $t.s$ that indicates the goodness (quality) of t and its domain of values is the range $[0, 1]$. Moreover, t is described by set of keywords $t.\mathcal{W}$ that capture the textual description of the feature object t. Figure 2.2 depicts an example of a set of feature objects that represent restaurants and shows the non-spatial score and the textual description. Table 2.1 provides an overview of the symbols used in this chapter.

Symbol	Description
O	Set of data objects
p	Data object, $p \in O$
c	Number of feature sets
F_i	Feature sets, $i \in [1, c]$
t	Feature object, $t \in F_i$
$t.s$	Non-spatial score of t
$t.\mathcal{W}$	Set of keywords of t
$dist(p, t)$	Distance between p and t
$sim(t, \mathcal{W})$	Textual similarity between t and $\mathcal{W}$
$s(t)$	Preference score of t
$\tau_i(p)$	Preference score of p based on F_i
$\tau(p)$	Spatio-textual preference score of p

Table 2.1: Overview of symbols.

The goal is to find data objects that have in their vicinity feature objects that (i) are of high quality and (ii) are relevant to the query keywords posed by the user. Thus, the score of the feature object t captures not only the non-spatial score of the feature, but its textual similarity to a user specified set of query keywords.

Definition 1 *The **preference score** $s(t)$ **of feature object** t based on a user-specified set of keywords $\mathcal{W}$ is defined as $s(t) = (1-\lambda) \cdot t.s + \lambda \cdot sim(t, \mathcal{W})$, where $\lambda \in [0, 1]$ and $sim()$ is a textual similarity function.*

The textual similarity between the keywords of the feature and the set $\mathcal{W}$ is measured by $sim(t, \mathcal{W})$ and its domain of values is the range $[0, 1]$. The parameter

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

λ is the smoothing parameter that determines how much the score of the feature objects should be influenced by the textual information. For the rest of the chapter, we assume that the textual similarity is equal to the Jaccard similarity between the keywords of the feature objects and the user-specified keywords: $sim(t, \mathcal{W}) = \frac{|t.\mathcal{W} \bigcap \mathcal{W}|}{|t.\mathcal{W} \bigcup \mathcal{W}|}$.

For example, consider the restaurants depicted in Figure 2.2. Given a set of keywords $\mathcal{W} = \{italian,\ pizza\}$ and $\lambda = 0.5$ the restaurant with the highest preference score is *Ontario's Pizza* with a preference score $s(r_6) = 0.9$, while the score of *Beijing Restaurant* is $s(r_1) = 0.3$, since none of the given keywords are included in the description of *Beijing Restaurant*.

Given a spatio-textual preference query Q defined by an integer k, a range r and c-sets of keywords $\mathcal{W}_i$, the preference score of a data object $p \in O$ based on a feature set F_i is defined by the scores of feature objects $t \in F_i$ in its spatial neighborhood, whereas the overall spatio-textual score of p is defined by taking into account all feature sets F_i, $1 \leq i \leq c$.

Definition 2 *The **preference score** $\tau_i(p)$ **of data object** p based on the feature set F_i is defined as:* $\tau_i(p) = max\{s(t) \mid t \in F_i : dist(p,t) \leq r \text{ and } sim(t, \mathcal{W}_i) > 0\}$.

The $dist(p,t)$ denotes the spatial distance between data object p and feature object t and we employ the Euclidean distance function. Continuing the previous example, Figure 2.4 shows the spatial location of the restaurants in Figure 2.2 and a data point p that represents a hotel. The preference score of p based on the restaurants in its neighborhood (assuming $r = 3.5$ and $\mathcal{W} = \{italian,\ pizza\}$) is equal to the score of r_6 ($\tau_i(p) = s(r_6) = 0.9$), which is the best restaurant in the neighborhood of p.

Definition 3 *The overall **spatio-textual preference score** $\tau(p)$ **of data object** p is defined as:* $\tau(p) = \sum_{i \in [1,c]} \tau_i(p)$.

Figure 2.3 shows a second set of feature objects that represents coffeehouses. For a tourist that looks for a good hotel that has nearby a good Italian restaurant that serves pizza and a good coffeehouse that serves espresso and muffins, the score of p would be $\tau(p) = s(r_6) + s(c_5) = 0.9 + 0.78233 = 1.6833$.

Problem 1 Top-k Spatio-Textual Preference Queries (STPQ)*: Given a query Q, defined by an integer k, a radius r and c-sets of keywords $\mathcal{W}_i$, find the k data objects $p \in O$ with the highest spatio-textual score $\tau(p)$.*

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

	name	rating	x	y	textual description
r_1	Beijing Restaurant	0.6	1	2	Chinese, Asian
r_2	Daphne's Restaurant	0.5	4	1	Greek, Mediterranean
r_3	Espanol Restaurant	0.8	5	8	Italian, Spanish, European
r_4	Golden Wok	0.8	2	3	Chinese, Buffet
r_5	John's Pizza Plaza	0.9	8	4	Pizza, Sandwiches, Subs
r_6	Ontario's Pizza	0.8	7	6	Pizza, Italian
r_7	Oyster House	0.8	6	10	Seafood, Mediterranean
r_8	Small Bistro	1.0	3	7	American, Coffee, Tea, Bistro

Figure 2.2: Feature objects (Restaurants)

	name	rating	x	y	textual description
c_1	Bakery & Cafe	0.6	4	1	Cake, Bread, Pastries
c_2	Coffee House	0.5	4	7	Cappuccino,Toast, Decaf
c_3	Coffe Time	0.8	3	10	Cake, Toast, Donuts
c_4	Cafe Ole	0.6	6	2	Cappuccino, Iced Coffee, Tea
c_5	Royal Coffe Shop	0.9	5	5	Muffins, Croissants,Espresso
c_6	Mocha Coffe House	1.0	10	3	Macchiato, Espresso, Decaf
c_7	The Terrace	0.7	6	9	Muffins, Pastries, Espresso
c_8	Espresso Bar	0.4	7	6	Croissants, Decaf, Tea

Figure 2.3: Feature objects (Coffeehouses)

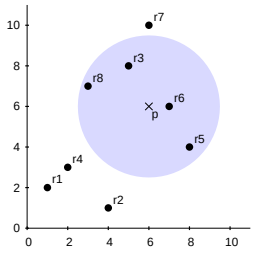


Figure 2.4: An example of a $STPQ$ query.

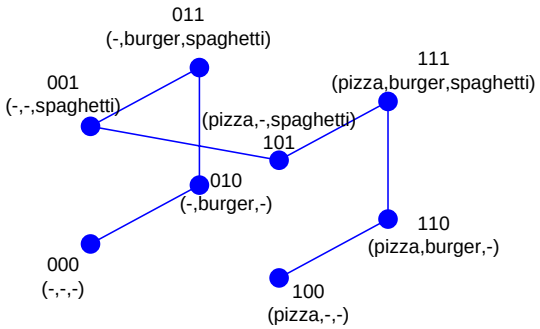


Figure 2.5: Hilbert-based keyword ordering.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

2.4 Indexing

The main difference of top-k spatio-textual preference queries to traditional spatio-textual search is that the ranking of a data object does not depend only on spatial location and textual information, but also on the non-spatial score of the feature object. In particular, the preference score $s(t)$ of feature object t is defined by its textual description and its non-spatial score, while the spatial location is used as a filter for computing the preference score $\tau_i(p)$ of data object p. Thus, efficient indexing of the textual description and the non-spatial score of feature objects is a significant factor for designing efficient algorithms for the STPQ query.

2.4.1 Index Characteristics

We assume that the data objects O are indexed by an R-Tree, denoted as *rtree*. However, for the feature objects, it is important that the non-spatial score and the textual description are indexed additionally. Each dataset F_i can be indexed by any spatio-textual index that relies on a spatial hierarchical index (such as the R-Tree). However, each entry e of the index must in addition maintain: (i) the maximum value of $t.s$ of any feature object t in the sub-tree, denoted as $e.s$, and (ii) a summary ($e.\mathcal{W}$) of all keywords of any feature t in the sub-tree. To ensure correctness of our algorithms, there must exist an upper bound $\widehat{s}(e)$ such that for any t stored in the sub-tree rooted by the entry e it holds:

$$\widehat{s}(e) \geq s(t)$$

The above property guarantees that the preference score $s(t)$ of a feature object t is bounded by the bound $\widehat{s}(e)$ of its ancestor node e. The efficiency of the algorithms directly depends on the tightness of this bound. In turn, this depends on the similarity between the textual description and the non-spatial score of the features objects that are indexed in the same node.

In the following, we propose an indexing technique that leads to tight bounds since objects with similar textual information and non-spatial score are stored in the same node of the index.

2.4.2 Indexing based on Hilbert Mapping

Our indexing approach maps the textual description of feature objects to a value based on the Hilbert curve. Let w denote the number of distinct keywords in the

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

vocabulary, then for each feature t the keywords $t.\mathcal{W}$ can be represented as a binary vector of length w. For instance, assuming a vocabulary $\{pizza, burger, spaghetti\}$, we can use an active bit to declare the existence of the *"pizza"* keyword at the first place, *"burger"* at the second, and *"spaghetti"* at the last. Moreover, we suggest a mapping of the binary vector to a Hilbert value, denoted as $\mathcal{H}(t.\mathcal{W})$. For the above w=3 keywords, the defined order is 000,010,011,001,101,111,110 and 100. Figure 2.5 shows the ordering of the keywords based on the Hilbert values. The benefit of this order is that it ensures us that vectors with distance 1 have only one different keyword, while if the distance is w', then the maximum number of different keywords is bound by w'. This means that consecutive vectors in the afore-described order have only few different keywords, which means that objects with sequential $\mathcal{H}$-values are highly similar also based on the Jaccard similarity function.

Using the Hilbert mapping of the textual information, each feature object t can be represented as a point in the 4-dimensional space $\{t.x, t.y, t.s, \mathcal{H}(t.\mathcal{W})\}$. Our indexing technique, called *SRT-index*, uses a spatial index, such as a traditional R-Tree, that is built on the mapped 4-dimensional space. In terms of structure, the SRT-index resembles a traditional R-Tree that it is built on the spatial location, the non-spatial score (rating), and the Hilbert value of the keywords of the feature objects altogether. The only modification needed during the index construction is the method used for updating the Hilbert values of a node. When the Hilbert value of a node is updated because a new object is added, then the previous Hilbert value as well the Hilbert value of the new object are mapped to binary vectors, the disjunction of the binary vectors is computed, mapped to a new Hilbert value and stored in the node. Notably, the exact spatial index used for indexing the mapped space does not affect the correctness of our algorithms, but only their performance. In our experimental evaluation, we use bulk insertion [25] on our novel indexing technique.

During query processing the bound $\widehat{s}(e)$ of a node e can be set as:

$$\widehat{s}(e) = (1 - \lambda) \cdot e.s + \lambda \cdot \frac{|e.\mathcal{W} \bigcap \mathcal{W}|}{|\mathcal{W}|}$$

where $\mathcal{W}$ is the set of query keywords, while $e.\mathcal{W}$ is the set of all keywords of all feature objects t indexed by the node e. The set $e.\mathcal{W}$ is computed based on the Hilbert mapping and the aggregated Hilbert value $\mathcal{H}(e.\mathcal{W})$ stored in the node entry e of the SRT-tree. It holds that $\widehat{s}(e) \geq s(t)$.

To summarize, the SRT-index overcomes the difficulty that other indexing approaches face, being unable to identify in advance what are the branches of the

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

index that store highly ranked and relevant feature objects to the query. The reason is that this indexing mechanism can identify effectively the promising parts of the hierarchical structure at a low cost, since during the index construction the similarity of the spatial location, the non-spatial score, as well as the textual description are taken into account.

2.5 Spatio-Textual Data Scan (STDS)

Our baseline approach, called spatio-textual data scan (*STDS*), computes the spatio-textual score $\tau(p)$ of *each* data object $p \in O$ and then reports the k data objects with the highest score. Algorithm 1 shows the pseudocode of *STDS*.

In more detail, for a data object p, its score $\tau_i(p)$ for every feature set F_i is computed (lines 3-5). The details on this computation for range queries are described in Algorithm 2 that will be presented in the sequel. Interestingly, for some data objects p we can avoid computing $\tau_i(p)$ for some feature sets. This is feasible because we can determine early that some data objects cannot be in the result set R. To achieve this goal, we define a threshold τ which is the k-th highest score of any data object processed so far. In addition, we define an upper bound $\widehat{\tau}(p)$ for the spatio-textual preference score $\tau(p)$ of p, which does not require knowledge of the preference scores $\tau_i(p)$ for all feature sets F_i: $\widehat{\tau}(p) = \sum_{i \in [1,c]} \begin{cases} \tau_i(p), & if\ \tau_i(p)\ is\ known \\ 1, & otherwise \end{cases}$. The algorithm tests the upper bound $\widehat{\tau}$ based on the already computed $\tau_i(p)$ against the current threshold (line 6). If $\widehat{\tau}$ is smaller than the current threshold, the remaining score computations are avoided. After computing the score of p, we test whether it belongs to R (line 6). If this is case, the result set R is updated (line 7), by adding p to it and removing the data object with the lowest score (in case that $|R| > k$). Finally, if at least k data objects have already been added to R, we update the threshold based on the k-th highest score (line 9).

The remaining challenge is to compute efficiently the score based on the spatio-textual information of the feature objects. The goal is to reduce the number of disk accesses for retrieving feature objects that are necessary for computing the score of each element $p \in O$. Algorithm 2 shows the computation of preference score $\tau_i(p)$ for feature set F_i. First, the root entry is retrieved and inserted in a heap (line 1). The heap maintains the entries e sorted based on their values $\widehat{s}(e)$. In each iteration (lines 2-11), the entry e with the highest value $\widehat{s}(e)$ is processed, following a best-first approach. If e is a data point and within distance r from p

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Algorithm 1 *Spatio-Textual Data Scan (*STDS*)*

Input: Query $Q = (k, r, \{\mathcal{W}_i\})$
Output: Result set R sorted based on $\tau(p)$
$R = \emptyset$; $\tau = -1$ **foreach** $p \in O$ **do**
  for $i = 1 \dots c$ **do**
    if $\widehat{\tau}(p) > \tau$ **then**
      $\tau_i(p) = F_i.computeScore(Q, p)$
  if $\tau(p) > \tau$ **then**
    update(R) **if** $|R| \geq k$ **then**
      $\tau = k^{th}$ score
return R

(line 5), then the score $\tau_i(p)$ of p has been found and is returned (line 7). If e is not a data point, then we expand it only if it satisfies the query constraints (line 9). More detailed, if the minimum distance of e to p is smaller or equal to r and its textual similarity is larger than 0, e is expanded and its child entries are added to the heap (line 11). Otherwise, the entire sub-tree rooted at e can be safely pruned.

Algorithm 2 *Spatio-Textual Score Computation on* F_i *(*$computeScore(Q, p)$*)*

Input: Query Q, data object p
Output: Score $\tau_i(p)$
$heap$.push(F_i.root) **while** ***not*** $heap.isEmpty()$ **do**
  $e \leftarrow heap$.pop() **if** e *is a data object* **then**
    if $(dist(p, e) \leq r)$ **then**
      $\tau_i(p) = s(e)$ **return** $\tau_i(p)$
  else
    if $(mindist(p, e) \leq r)$ ***and*** $(sim(e, \mathcal{W}_i) > 0)$ **then**
      for *childEntry* ***in*** *e.childNodes* **do**
        $heap$.push(childEntry)

Correctness and Efficiency: Algorithm 2 always reports the correct score $\tau_i(p)$. The sorted access of the entries, combined with the property that the value $\widehat{s}(e)$ of the entry is an upper bound ensure its correctness. Moreover, it can be shown that Algorithm 2 expands the minimum number of entries, in the sense that if an entry that is expanded was not expanded, it could lead to computing a wrong score. This is because only entries with score higher than any processed feature object

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

are expanded, and such entries may contain in their sub-tree a feature object with score equal to the score of the entry.

Performance improvements: The performance of *STDS* can be improved by processing the score computations in a batch. Instead of a single data object p, a set of data objects $\mathcal{P}$ can be given as input to Algorithm 2. Then, an entry is expanded if the distance for *at least* one p in $\mathcal{P}$ is smaller than r. When a feature object is retrieved, for any p for which the distance is smaller than r the score is computed and those data objects p are removed from $\mathcal{P}$. The same procedure is followed until either the heap or $\mathcal{P}$ is empty. Algorithm 1 can be easily modified to invoke Algorithm 2 for all data objects in the same leaf entry of the R-tree (*rtree*) that indexes the data objects O. For sake of simplicity, we omit the implementation details, even though we use this improved modification in our experimental evaluation.

2.6 Spatio-Textual Preference Search (STPS)

In this section we propose a novel and efficient algorithm, called Spatio-Textual Preference Search (*STPS*), for processing spatio-textual preference queries. *STPS* follows a different strategy than *STDS*, as it involves two major steps, namely finding highly ranked feature objects first, and then, retrieving data objects in their spatial neighborhood. Intuitively, if we find a neighborhood in which highly ranked feature objects exist, then the neighboring data objects are naturally highly ranked as well.

2.6.1 Valid Combination of Feature Objects

In a nutshell, the goal is to find sets of feature objects $\mathcal{C} = \{t_1, t_2, \ldots, t_c\}$ where $t_i \in F_i$ $(1 \leq i \leq c)$, such that the spatio-textual preference score of each t_i is as high as possible and the feature objects are located in nearby locations.

In the general case, a data object may be highly ranked even in the case where a certain kind of feature object does not exist in its neighborhood, though feature objects of other kinds might compensate for this. For example, consider the extreme case where all data objects have only one type of feature object in their spatial neighborhood. For ease of presentation, we denote as $\emptyset$ a virtual feature object for which it holds that $dist(p, \emptyset) = 0$, $dist(t_i, \emptyset) = 0$ and $s(\emptyset) = 0$ $\forall t_i, p$. This virtual feature object is used for presenting unified definitions for the case where the spatio-textual score of the top-k data objects is defined based on less

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Algorithm 3 *Spatio-Textual Preference Search (STPS)*
Input: Query Q
Output: Result set R sorted based on $\tau(p)$
while *(*$|R| \leq k$*)* **do**
  $\mathcal{C} = nextCombination(Q)$ $R = R \cup$ getDataObjects($\mathcal{C}$)
return R

than c feature objects. More formally put, we define the concept of *valid combination* of feature objects as:

Definition 4 *A valid combination of feature objects is a set* $\mathcal{C} = \{t_1, t_2, \ldots, t_c\}$ *such that (i)* $\forall i\ t_i \in F_i$ *or* $t_i = \emptyset$*, and (ii)* $dist(t_i, t_j) \leq 2r\ \forall i, j$*. The score of the valid combination* $\mathcal{C}$ *is defined as* $s(\mathcal{C}) = \sum_{1 \leq i \leq c} s(t_i)$*.*

The following lemma proves that it is sufficient to examine only the valid combinations $\mathcal{C}$ of feature objects in order to retrieve the result set of a top-k spatio-textual preference query.

Lemma 1 *The score of any data object* $p \in O$ *is defined by a valid combination of feature objects* $\mathcal{C} = \{t_1, t_2, \ldots, t_c\}$*, i.e.,* $\forall p : \exists \mathcal{C} = \{t_1, t_2, \ldots, t_c\}$ *such that* $\tau(p) = s(\mathcal{C})$

Proof Let us assume that there exists p such that: $\tau(p) = \sum_{i \in [1,c]} \tau_i(p)$ with $\tau_i(p) = \{s(t_i) \mid t_i \in F_i : dist(p, t_i) \leq r \ and \ sim(t_i, \mathcal{W}_i) > 0\}$ and $\mathcal{C} = \{t_1, t_2, \ldots, t_c\}$ is not a valid combination of feature objects. Since $\mathcal{C} = \{t_1, t_2, \ldots, t_c\}$ is not a valid combination of feature objects, there exists $1 \leq i \neq j \leq c$ such that $dist(t_i, t_j) > 2r$ but also $dist(p, t_i) \leq r$ and $dist(p, t_j) \leq r$. Based on the triangular inequality it holds: $dist(t_i, t_j) \leq dist(p, t_i) + dist(p, t_j) \leq r + r \leq 2r$, which is a contradiction.

2.6.2 STPS Overview

Algorithm 3 provides an insight to *STPS* algorithm. At each iteration, the following steps are followed: (i) a special iterator (line 2) returns successively the valid combinations of feature objects sorted based on their score (we discuss the details on the implementation of the iterator in the following subsection), (ii) up to k data points in the spatial neighborhood of these features are retrieved (line

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

3). Data objects that have already been previously retrieved are discarded, while the remaining data objects p have a score $\tau(p) = s(\mathcal{C})$ and can be returned to the user incrementally. If k data objects have been returned to the user (line 1), the algorithm terminates without retrieving the remaining combinations of feature objects. Differently to the *STDS* algorithm, *STPS* retrieves only the data objects that most certainly belong to the result set.

Algorithm 4 *Spatio-Textual Feature Objects Retrieval* ($nextCombination(Q)$)

Input: Query Q
$heap_i$: heap maintaining entries of F_i
$heap$: heap maintaining valid combinations of feature objects
$\mathcal{D}_i$: set of feature objects of F_i
Output: $\mathcal{C}$: valid combination with highest score
while *(*$\exists i$: ***not*** $heap_i$*.isEmpty())* **do**
  $i \leftarrow$ nextFeatureSet() $e_i \leftarrow heap_i.pop()$ **while** *(**not** e_i is a data object)* **do**
    for *childEntry **in** e_i.childNodes* **do**
      $heap_i$.push(childEntry)
    $e_i \leftarrow heap_i$.pop()
  $\mathcal{D}_i = \mathcal{D}_i \cup e_i$ $heap$.push(validCombinations($\mathcal{D}_1, \cdots, e_i, \cdots, \mathcal{D}_c$)) $min_i = s(e_i)$ $\tau = max_{1 \leq j \leq c}(max_1 + \cdots + min_j + \cdots + max_c)$ $\mathcal{C} \leftarrow heap$.top()
  if *(score($\mathcal{C}$)* $\geq \tau$*)* **then**
    $heap$.pop() **return** $\mathcal{C}$

2.6.3 Spatio-Textual Feature Objects Retrieval

Algorithm 4 shows the pseudocode for retrieving the valid combinations $\mathcal{C} = \{t_1, t_2, \ldots, t_c\}$ of feature objects sorted based on their spatio-textual preference score $s(\mathcal{C})$. We first give a scketch of our algorithm and then we will elaborate further on the details in the following of this section. In each iteration, a feature set F_i is selected (line 2) based on a pulling strategy implemented by $nextFeatureSet()$. The spatio-textual index that stores the feature objects of the feature set F_i is accessed and the feature objects t_i are retrieved based on their score $s(t_i)$ that aggregates their non-spatial score, but also their textual similarity to the query keywords (lines 3-7). The retrieved feature objects are maintained in a list $\mathcal{D}_i$ (line 8) and are used to produce valid combinations $\mathcal{C}$ of feature objects (line 9). Moreover, a thresholding scheme is employed to decide when the combination with the highest score has been retrieved (lines 11-15).

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

We denote as max_i the maximum score of $\mathcal{D}_i$ and min_i the minimum score of $\mathcal{D}_i$. Thus, min_i represents the best potential score of any feature object of F_i that has not been processed yet. Moreover, in Algorithm 4 the variables $heap_i$, $\mathcal{D}_i$, max_i, min_i, and $heap$ are global variables. They are initialized as following $heap_i$: the root of F_i, $\mathcal{D}_i = \emptyset$ and $heap = \emptyset$, $min_i = \infty$. Variable max_i is the score of the highest ranked feature object of F_i and is set the first time the F_i index is accessed.

Accessing F_i: In each iteration, Algorithm 4 accesses one spatio-textual index that stores the set F_i (lines 3-7). The entries of the spatio-textual index responsible for the feature objects of F_i are maintained in $heap_i$, which keeps the entries e sorted based on $\widehat{s}(e)$. Moreover, for sake of simplicity, we assume that $heap_i.pop()$ will return a virtual feature object $t_i = \emptyset$ (with score equal to 0) as final object. In each iteration an entry e_i of the spatio-textual index is retrieved from $heap_i$ (line 3). If the entry e_i corresponds to a node of the index, the entry is expanded and its child nodes are added to the $heap_i$ (lines 5-6). Algorithm 4 continues retrieving from $heap_i$ entries, until an entry that is a feature object is retrieved (line 4). When an entry e_i is retrieved that corresponds to a feature object, e_i is inserted in the list $\mathcal{D}_i$ (line 8).

Creation of $\mathcal{C}$: After retrieving a new feature object e_i, new valid combinations $\mathcal{C}$ are created by combining e_i with the previously retrieved feature objects t_j maintained in the lists $\mathcal{D}_j$ (line 9). For this, the method $validCombinations$ is called, which returns all combinations of the objects in $\mathcal{D}_j$ and e_i, by discarding combinations for which the condition $dist(t_i, t_j) \leq 2r \ \forall i, j$ does not hold. The new valid combinations are inserted in the $heap$ (line 9) that maintains the valid combinations sorted based on their score $s(\mathcal{C})$.

Thresholding scheme: Algorithm 4 employs a thresholding scheme to determine if the current best valid combination can be returned as the valid combination with the highest score. The threshold τ represents the best score of any valid combination of feature objects that has not been examined yet. The best score of the next feature object t_j retrieved from F_j is equal to min_j, since the feature objects are accessed sorted based on $s(t_j)$. Obviously, for the remaining feature sets we assume that the new feature object t_j is combined with the feature objects that have the highest score. Thus, $\tau = max_{1 \leq j \leq c}(max_1 + \cdots + min_j + \cdots + max_c)$ (line 11) is an upper bound of the score for any valid combination that has not been examined yet. In line 13, we test whether the best combination of feature objects in the $heap$ has a score higher or equal to the threshold τ. If so, the best combination in the heap is the next valid combination with the best score. Otherwise, additional feature objects from feature sets F_i have to be retrieved

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

until it holds that the top element of the $heap$ achieves a score which is higher than τ.

Pulling strategy: In the following, we proposed an advanced pulling strategy that prioritizes retrieval from feature sets that have higher potential to produce the next valid combination $\mathcal{C}$. A simple alternative would be to access the different feature sets in a round robin fashion.

The order in which the feature objects of different feature sets are retrieved is defined by a pulling strategy, i.e., $nextFeatureSet()$ returns an integer between 1 and c and defines the pulling strategy. In addition, $nextFeatureSet()$ never returns i if $heap_i$ is empty.

Definition 5 *Given c sets of feature objects $\mathcal{D}_i$, the prioritized pulling strategy returns m as the next feature set such that $\tau = max_1 + \cdots + min_m + \cdots + max_c$.*

The main idea of the prioritized pulling strategy is that in each iteration the feature set F_m that is responsible for the threshold value τ is accessed. It is obvious that the only way to reduce τ is to reduce the min_m, since retrieval from the remaining feature sets cannot reduce τ. Thus, retrieving the next tuple from the feature set F_m may reduce the threshold τ and may produce new valid combinations that have a score equal to the current threshold.

2.6.4 Retrieval of Qualified Data Objects

In the following, we study the reciprocal actions taken upon the formation of a highly ranked combination of feature objects.

In Algorithm 3 (line 3) *getObjects($\mathcal{C}$)* is invoked to retrieve from $rtree$ all data objects in the neighborhood of the feature objects in $\mathcal{C}$. This method starts from the root of the $rtree$ and processes its entries recursively. Entries e for which $\exists i$ such that $t_i \in \mathcal{C}$ with $dist(e, t_i) > r$ are discarded. The remaining entries are expanded until all objects p for which it holds that $dist(p, t_i) \leq r$ are retrieved.

Example. *Consider for example the feature sets depicted in Figure 2.2 and in Figure 2.3. Given a query with $r = 3.5$, $\mathcal{W}_1 = \{italian,\ pizza\}$ and $\mathcal{W}_2 = \{espresso,\ muffins\}$, the restaurant and the coffeehouse with the highest scores are r_6 and c_5 respectively. Since it holds that $dist(r_6, c_5) \leq 2r$, the set $\mathcal{C} = \{r_6, c_5\}$ is a valid combination of feature objects. Assume that the set of data objects is $O = \{p_1, p_2, \ldots, p_{10}\}$ as depicted in Figure 2.6. For the data objects p_6, p_9 and p_{10} it holds that $dist(p_i, c_5) \leq r$ and $dist(p_i, r_6) \leq r$, and their spatial-textual score is $\tau(p_6) = \tau(p_9) = \tau(p_{10}) = 1.6833$. These data objects are guaranteed to*

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

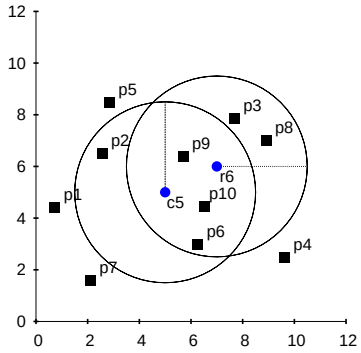


Figure 2.6: Data objects within qualifying distance from $\mathcal{C} = \{r_6, c_5\}$.

be the highest ranked data objects and can be immediately returned to the user. For $k \leq 3$, our algorithm terminates without examining other feature combinations.

2.7 Variants of Top-k Spatio-Textual Preference Queries

In this section, we extend our algorithms for processing spatio-textual preference queries based on alternative score definitions under a unified framework. We provide formal definitions for the alternative score definitions, namely *influence preference score* and *nearest neighbor preference score*. Moreover, we discuss for all query types the necessary modifications to our query processing algorithms.

2.7.1 Influence-Based STPQ Queries

In contrast to the preference score defined in Definition 1 (in the following referred to as range score), in this section we define an alternative score that does not pose a hard constraint on the distance, but instead gradually reduces the score based on the distance. We call this variant *influence preference score*.

Definition 6 *The **influence preference score** $\tau_i(p)$ of data object p based on the feature set F_i is defined as: $\tau_i(p) = max\{s(t) \cdot 2^{\frac{-dist(p,t)}{r}} \mid t \in F_i : \ sim(t, \mathcal{W}_i) > 0\}$.*

The overall spatio-textual score $\tau(p)$ of data object p is defined as in the case of the range score, and the query returns the k objects with the highest score.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

The *STDS* algorithm, as defined in Algorithm 1 can be easily adapted for the case of influence score. Only the function $computeScore(Q, p)$ must be modified according to the definition of the score variant. Thus, in Algorithm 2 each entry in line 2.5 will be prioritized according to the influence preference score. In addition, the range restriction is removed in line 5 and line 9. No further modifications are needed, thus in the following we focus on the modifications and optimizations needed for *STPS* algorithm.

Algorithm 5 *STPS for influence score*

Input: Query Q
Output: Result set R sorted based on $\tau(p)$
$\tau = 0$ $score = -1$ **while** *(*$|R| \leq k$*)* ***or*** *(best ¿* τ*)* **do**
  $\mathcal{C} = nextCombination(Q)$ $best = s(\mathcal{C})$ $R = R \cup$ getDataObjects($\mathcal{C}$) $\tau =$ k-th score in R
return R

$STPQ$ queries based on the influence preference score can be efficiently supported by the *STPS* algorithm with few modifications. Algorithm 5 shows the modified *STPS* for influence preference score. The algorithm continues until at least k data object have been retrieved and until we are sure that none of the remaining data objects can have a better score. We use the score of the k-th data object of the current top-k result (line 7) to set a threshold τ. Hence, if the $best$ score of any unseen combination is smaller or equal to τ, the algorithm naturally terminates. In more details, $\mathcal{C} = nextCombination(Q)$ is the same with Algorithm 4 and returns the best combination based on score $s(\mathcal{C})$, but without discarding combinations whose distance is greater than $2r$. Thus, in each iteration the combination $\mathcal{C}$ with the highest $\tau(p) = \sum_{i \in [1,c]} \tau_i(p)$ is retrieved. Recall that for the case of the range preference score, all data objects that were located in distance smaller than r from all feature objects of $\mathcal{C}$ had a score equal to $s(\mathcal{C})$. Instead in the case of the influence preference score, $s(\mathcal{C})$ is an upper bound for the score of all data objects based on $\mathcal{C}$. This is because, the computed score is the influence score only for the objects with distance 0, while all other objects have a smaller influence score. Therefore, *getDataObjects*($\mathcal{C}$) must be modified accordingly.

In more details, $getDataObjects()$ retrieves the k points that have the highest influence score, by starting a top-k query on the R-Tree (*rtree*) of the data objects. The root is inserted in a heap sorted by the influence score ($\tau(p) = \sum_{i \in [1,c]} \tau_i(p) \dot{2}^{\frac{-dist(p,t_i)}{r}}$). For non-leaf entries e the influence score is computed

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

based on the mindist. Then, the influence score of an entry is an upper bound of any object in the subtree. After retrieving k data objects, we have retrieved the k data objects with the highest influence score for this combination of feature objects. Further improvement can be achieved if $getDataObjects()$ stops retrieving data objects based on τ, which reduces the I/Os on *rtree*. If τ is given to $getDataObjects()$ then it will return at most k data objects that have a score smaller than τ. Line 6 merges the results while it removes objects that have been retrieved before. Thus, if an object that is already in the heap is retrieved again the score with the highest value is kept.

After retrieving k data objects with the highest $\tau(p)$ in line 6 (Algorithm 5), the score of the k-th data object in R is used as a threshold τ (line 7). The best score of any unseen combination is $best = s(\mathcal{C})$, which is also an upper bound for the score of any unseen data object, since this is the score for distance 0. Hence, if the $best$ score is greater than τ, we have to retrieve additional objects. If the score $s(\mathcal{C})$ of the next combination is smaller than or equal to the threshold we stop retrieving other combinations.

2.7.2 Nearest Neighbor STPQ Queries

In the next score variant, each data object takes as a score the goodness of the feature objects that are its nearest neighbors. In particular, for each feature set the score of the nearest feature object is considered for computing the score of a data object.

Definition 7 *The **nearest neighbor preference score** $\tau_i(p)$ **of data object** p based on the feature set F_i is defined as:* $\tau_i(p) = \{s(t) \mid t \in F_i : dist(p,t) \leq dist(p,t') \; \forall t' \in F_i \; and \; sim(t,\mathcal{W}_i) > 0\}$

The overall spatio-textual score $\tau(p)$ of data object p is defined as in the case of the range score, and the query returns the k objects with the highest score. Again, *STDS* treats nearest neighbor queries similarly as in Algorithm 2 with subtle changes. The range predicate is removed in line 5 and line 9, while the child entries are prioritized in the heap according to their minimum distance from the data object p.

Regarding *STPS*, Algorithm 3 is directly applicable for the nearest neighbor score by modifying $nextCombination(Q)$ of Algorithm 4 to return the best combination based on score $s()$, but without discarding combinations that have a

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

$distance > 2r$, as also in the case of the influence score. The remaining challenge is given a combination $\mathcal{C}$ to retrieve the data objects that satisfy the nearest neighbor requirement.

Generally, it is more difficult compared to the other score variants to retrieve the data objects for a given combination $\mathcal{C}$. We need to retrieve all data objects for which the nearest neighbor t_i based on F_i belongs to $\mathcal{C}$. For each feature object t_i of $\mathcal{C}$, there exists a region in which all data points that fall into that region have t_i as their nearest neighbor. This region corresponds to the Voronoi cell [34] and this problem has been studied for finding reverse nearest neighbors [26]. Only the data objects in the intersection of all regions need to be retrieved. In fact, we compute incrementally the Voronoi cell for each feature object t_i of $\mathcal{C}$, which allows us to discard early combinations for which the intersection becomes empty. We omit further implementation details due to space limitations.

2.8 Experimental Evaluation

In this section, we evaluate the performance of our algorithms *STDS* and *STPS*, presented previously in Section 2.5 and Section 2.6 respectively, for processing spatio-textual preference queries over large disk-resident data. Moreover, we study the gains in performance of our algorithms caused by the SRT index proposed in Section 2.4 compared to an existing indexing technique (IR^2-Tree [17]). In order to ensure a fair comparison, we modify the IR^2-Tree to support score values of feature objects. To this end, we add to the leaf nodes of IR^2-Tree the scoring values for the feature objects, and maintain in ancestor (internal) nodes the maximum score of all enclosed feature objects. All experiments run on an Intel 2.2GHz processor equipped with 2GB RAM.

2.8.1 Experimental Setup

Methodology. In our experimental evaluation, we vary four important parameters of the datasets in order to study the scalability of the proposed techniques (Section 2.8.2). These parameters are: (i) the cardinality of the feature sets $|F_i|$, (ii) the cardinality of the set of data objects $|O|$, (iii) the number of feature sets c, and (iv) number of distinct keywords indexed. Moreover, we study four different query parameters to study how the characteristics of the query influence the performance of the algorithms (Section 2.8.3). In more details, we vary (i) the query radius r, (ii) the number k of retrieved data objects, (iii) the smoothing pa-

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

rameter λ between textual similarity and non-spatial score, and (iv) the number of keywords of the query for each feature set. Finally, we evaluate the performance of *STPS* for the influence score variant (Section 2.8.4) as well as for the nearest neighbor variant (Section 2.8.5).

Tested ranges for all parameters are shown in Table 2.2. The default values are denoted as bold. When we vary one parameter, all others are set to their default values.

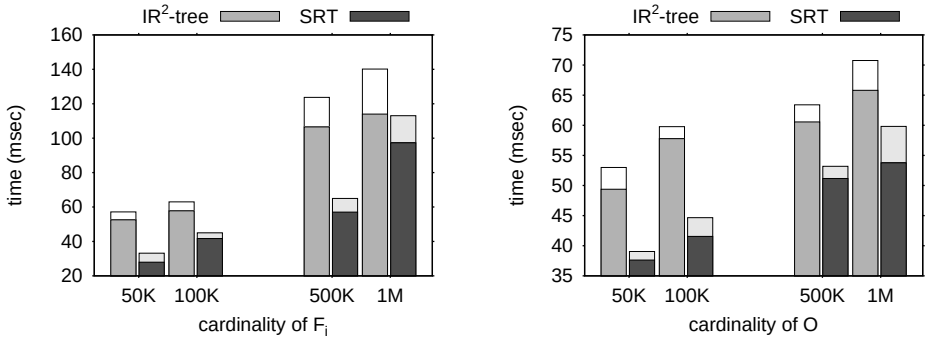
Parameter	**Range**
Cardinality of dataset	$50K$, **100K**, $500K$, $1M$
Cardinality of features sets	$50K$, **100K**, $500K$, $1M$
Number of feature sets c	**2**, 3, 4, 5
Indexed keywords	64, **128**, 192, 256
Radius r (norm. in $[0, 1]$)	.005, **.01**, .02, .04, .08
k	5, **10**, 20, 40, 80
Smoothing parameter	.1, .3, **.5**, .7, .9
Queried keywords	1, **3**, 5, 7, 9

Table 2.2: Experimental parameters.

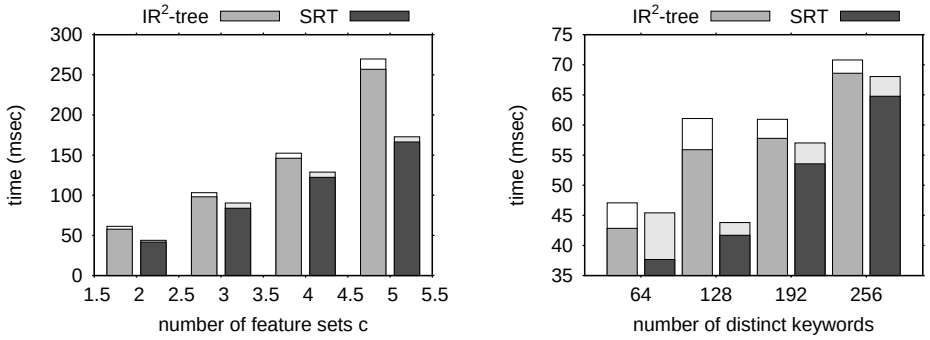
Datasets. For evaluating our algorithms, we use both real and synthetic datasets. The real dataset, which was obtained from `factual.com`, describes hotels ($\approx$ 25K objects) and restaurants ($\approx$ 79K objects). In more details we collected restaurant and hotels that are annotated with their location. Moreover, for the collected restaurants we extracted their rating and their textual description of the served food, mentioned as "cuisine". The number of distinct values of keywords for the cuisine is around 130 and each restaurant description may contain one or more keywords. Our datasets contain collected hotels and restaurants for 13 US states that are the states for which `factual.com` lists sufficient data. In addition, we created synthetic clustered datasets of varying size, number of keywords and number of feature sets. Approximately $10,000$ clusters constitute each synthetic dataset. The number of distinct keywords is set to 256 as a default value and each feature object is characterized by one or more keywords that are picked randomly. The spatial constituent of all datasets has been normalized in $[0, 1] \times [0, 1]$. Every reported value is the average of $1,000$ random queries, which are generated in a similar way as the synthetic data and follow the same data distribution.

Metrics. The efficiency of all schemes is evaluated according to the average execution time required by a query (time consumed in the CPU and to read disk-pages). In our figures we break down the execution time into the time consumed

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) Cardinality of F_i (b) Cardinality of O



(c) Varying c (d) Distinct keywords

Figure 2.7: Scalability for synthetic dataset.

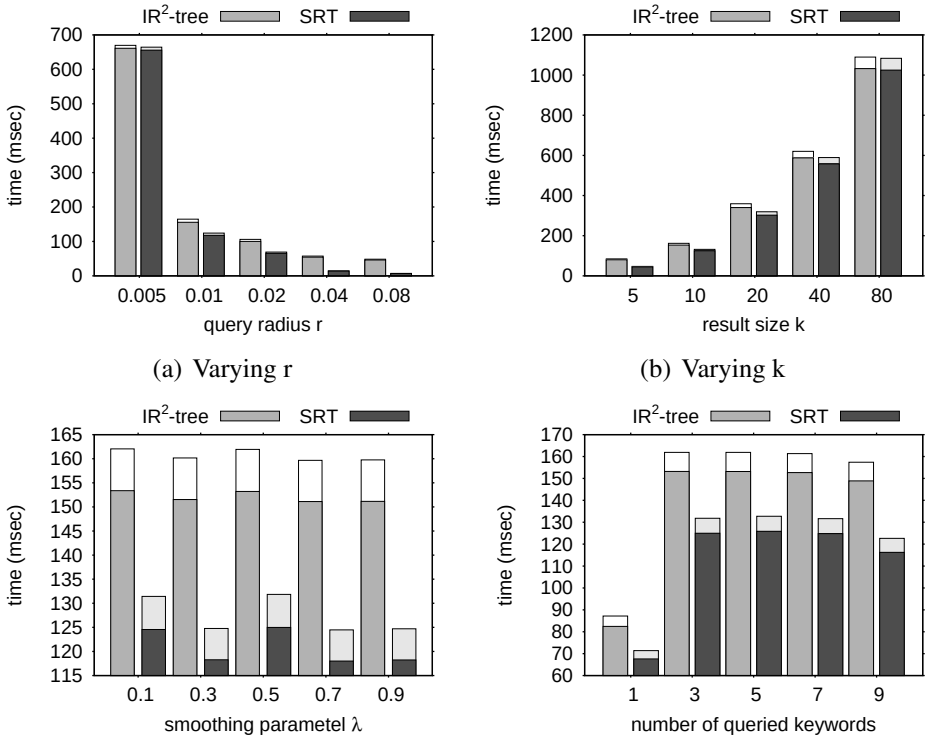
due to the disk accesses (dark part of the bars) and the time needed for processing the query (CPU time) which is the white part of the bars. The time consumed due to the disk accesses relates to the number of the required I/Os.

2.8.2 Scalability Analysis

In this section, we evaluate the impact of varying different parameters on the efficiency of our algorithms. In order to perform a scalability analysis, we employ the synthetic dataset for this set of experiments. First, we show the scalability limitations of *STDS* for large datasets (Table 2.3), and then we explore in more detail the significantly superior performance of *STPS*.

Table 2.3 shows the results for *STDS* when varying different parameters of the dataset. For the default setting, *STDS* requires over 13 seconds for range queries. Evidently, when a large number of data objects is involved *STDS* does

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) Varying r (b) Varying k

(c) Varying λ (d) Queried keywords

Figure 2.8: Range query parameters for real dataset.

not scale well and the absolute time required is high. The main reason is that *STDS* associates all data objects with c feature objects, which is particularly time-consuming. This experiment demonstrates that a plain algorithm for solving the problem can lead to prohibitive processing cost. Since *STDS* performs badly for all experimental setups, we omit *STDS* for the rest of experimental evaluation, and study the performance of *STPS* coupled with two different indexing techniques.

Figure 2.7 illustrates the results for the same experiment as above, but for the *STPS* algorithm. We implemented *STPS* over two different indexes: (i) our SRT index (proposed in Section 2.4), and (ii) the modified IR^2-Tree [17] whose nodes are enhanced with the maximum score of enclosed feature objects. In summary, the results clearly demonstrate that *STPS* scales with all parameters and that SRT indexing always outperforms IR^2-tree. Moreover, in both cases, the *STPS* algorithm exhibits high performance, as witnessed by the low execution time, which stems from its ability to quickly identify qualified feature combinations. Conse-

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union

Feature objects $\|F_i\|$	50000	100000	500000	1000000
IR^2-tree	13427.3	13854.6	25223.1	31434.6
SRT	12301.7	13187.9	18725.1	23046.3
Data objects $\|O\|$	50000	100000	500000	1000000
IR^2-tree	13073.2	13854.6	21074.2	27846.0
SRT	11718.1	13187.9	18267.4	23444.9
Number c **of** F_i	2	3	4	5
IR^2-tree	13854.6	27842.6	33625.0	40188.4
SRT	13187.9	14104.9	32071.1	38340.7
Indexed keywords	1	2	3	4
IR^2-tree	13698.7	13854.6	15655.6	16209.6
SRT	13121.4	13187.9	13207.9	13887.8

Table 2.3: *STDS* execution time (in msec) for synthetic dataset.

quently, the significant gains in processing time (orders of magnitude compared to *STDS*) are mostly due to the effective design of the algorithm. The SRT index additionally offers a speedup of x2 compared to the IR^2-Tree, which further improves the overall performance.

Figure 2.7(a) shows the execution time when increasing the cardinality of the feature sets. *STPS* scales well since the execution time increases only by a factor of at most x3, when increasing the dataset by one order of magnitude. This increase is due to the increased size of the data structures and the additional processing required to traverse a bigger data structure and find valid combinations of high score. When comparing the index structures, the SRT index is faster, due to the clustering of all score constituents (distance, textual similarity, and non-spatial score) in the 4-dimensional space.

Figure 2.7(b) shows the obtained results when increasing the number of data objects. Again, *STPS* scales well, and, in fact, even better than in the previous experiment. Obviously, a larger dataset of data objects does not affect the performance so much as larger feature sets. Again, the use of SRT indexing consistently outperforms the IR^2-Tree.

In Figure 2.7(c), we increase the number of feature categories c. As expected, this has a stronger effect on performance, since the cost required to retrieve the highest ranked combinations increases with the number of possible combinations, which, in turn, increases exponentially with c. Still, the performance of *STPS* is not severely affected, especially in the case of the SRT index which scales gracefully with c.

In Figure 2.7(d), we illustrate how the performance is affected by the number of distinct keywords in the dataset. Apparently, a higher number of keywords causes higher execution times. The reason is twofold. First, as the number of

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

distinct keywords increases, it is less probable to find feature objects that are described by all queried keywords, thus more feature objects need to be retrieved in order to ensure that no other combination has a higher score. Secondly, the node capacity of the index structures drops, thus the height of the index structures may increase, thus causing more IOs. In any case, the increase in the absolute value of execution time is relatively small (20 msec), even when we increase the vocabulary by a factor of 4 (from 64 to 256 keywords).

2.8.3 Varying Query Parameters

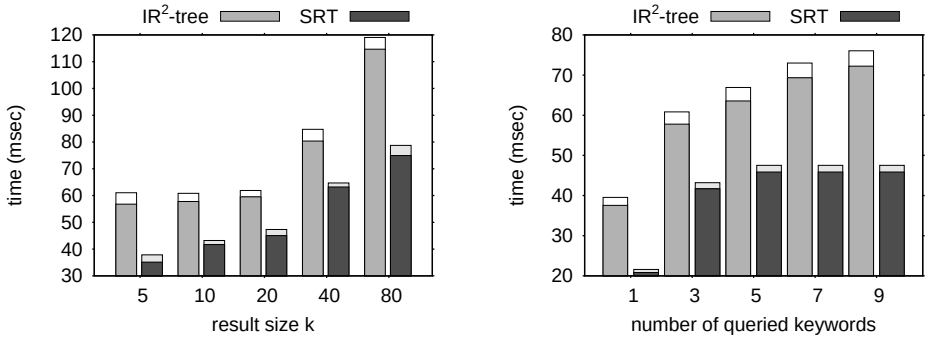
In Figure 2.8, we study the effect of varying query parameters for the real dataset. First, in Figure 2.8(a), we evaluate the impact of increasing the query radius r on the performance of *STPS*. We notice that for smaller values of r the execution time increases and the gain of SRT indexing compared to IR^2-tree drops. For small radius, access to more qualified combinations of feature objects is required, since only few data objects are located in their neighborhood. Therefore, for both indexing approaches the execution time increases mainly due to the increase of the IOs. Since a high percentage of the feature objects need to be retrieved for each feature set, the gain of SRT indexing is small. However, difference in performance becomes obvious for greater values of r, and hence, finding relevant feature objects in terms of textual description and good non-spatial score becomes most important for accessing only few feature objects.

Figure 2.8(b) illustrates the execution time when varying the size of result set k. Overall, the execution time increases as k increases. Specifically, with higher values of k more combinations of feature objects are retrieved to compose the result set, which again lead to more IOs to retrieve the qualifying feature objects that constitute valid combinations.

In Figure 2.8(c), we vary the smoothing parameter λ. In general, both approaches exhibit relatively stable performance for varying values of λ. The performance of IR^2-tree is not affected by the smoothing parameter, since the feature objects are not grouped into blocks based on the non-spatial score nor based on their textual similarity. We note for the IR^2-tree that objects with similar textual descriptions are stored throughout the index, regardless of their non-spatial score; unlike the SRT index where they are clustered together in the same block. As a result, a significant overhead is evident when searching for relevant objects all over the IR^2-tree. On the other hand, the SRT index is built by taking into account non-spatial score, the textual information and the spatial location. Thus, *STPS* that uses SRT index is consistently more efficient regardless of the value of the

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

smoothing parameter.



(a) Varying k (b) Queried keywords

Figure 2.9: Query parameters for synthetic dataset.

In Figure 2.8(d), we vary the number of queried keywords per feature set from 1 to 9. The number of queried keywords has little impact on performance, except for the special case where one keyword is queried for each feature set. This is because both of the indexing techniques aggregate in the non-leaf nodes the textual information of the leaf nodes, which makes it much easier to find objects that contain one keyword, rather than finding objects that are described with more keywords. Nevertheless, the gain in execution time of SRT indexing compared to the IR^2-tree is obvious.

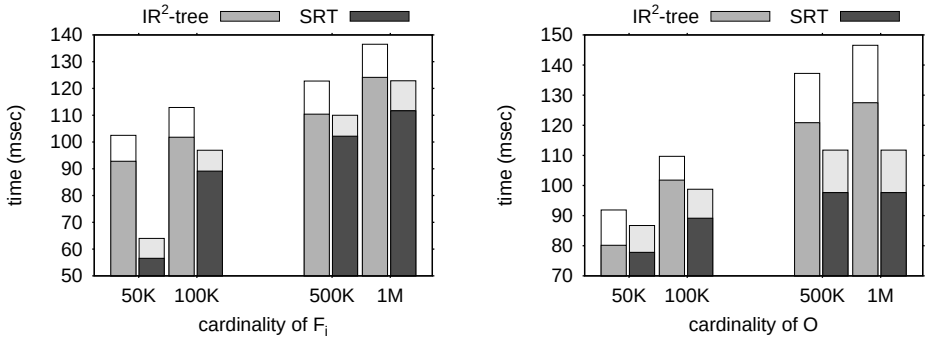
Figure 2.9 depicts results obtained from the synthetic dataset, when varying different query parameters. We notice the same tendency as in the case of the real dataset. In general, we observed that range queries are costlier for the real dataset. This is due to the data distribution: our real dataset, which was extracted from `factual.com`, consists of restaurants and hotels in the US forming just a few clusters. On the other hand, our synthetic dataset is substantially larger and contains a few thousands of clusters. Hence, the data from the latter dataset are more dispersed compared to the former. Last but not least, the SRT indexing consistently outperforms the IR^2-tree.

2.8.4 Influence-based Preference Score

In this section, we study the performance of *STPS* for the influence-based score variant of the spatio-textual preference queries. Figure 2.10 shows the scalability analysis of *STPS* for this query variant. By comparing the results to Figure 2.7,

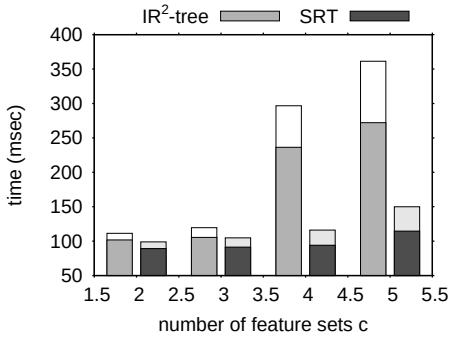
The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union

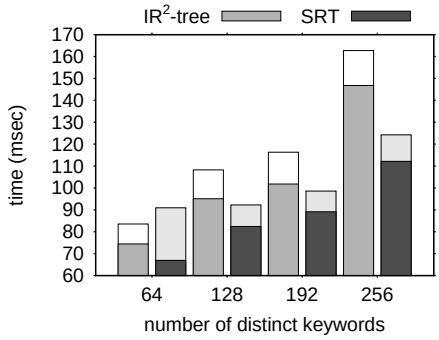


(a) Cardinality of F_i

(b) Cardinality of O



(c) Varying c



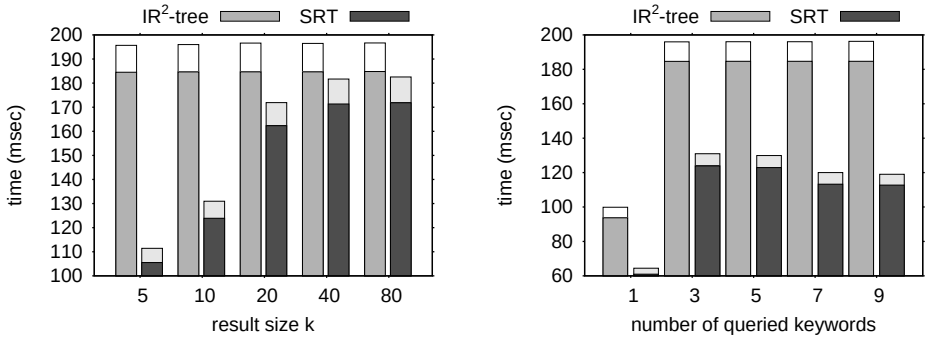
(d) Distinct keywords

Figure 2.10: Scalability for synthetic dataset and infuence queries.

which studies the execution time of the range score variant for the same parameters, we conclude that the required execution time is comparable and in some cases slightly increased. This is because more data object for each combination must be retrieved (for the influence-based score variant), since data objects that are further away than r may also have a non-zero score. Nevertheless, the additional cost is not significant in our experiments, and we notice the same tendency in execution time as in the case of range score, thus similar conclusions can be drawn. Moreover, the SRT indexing technique is beneficial in all setups.

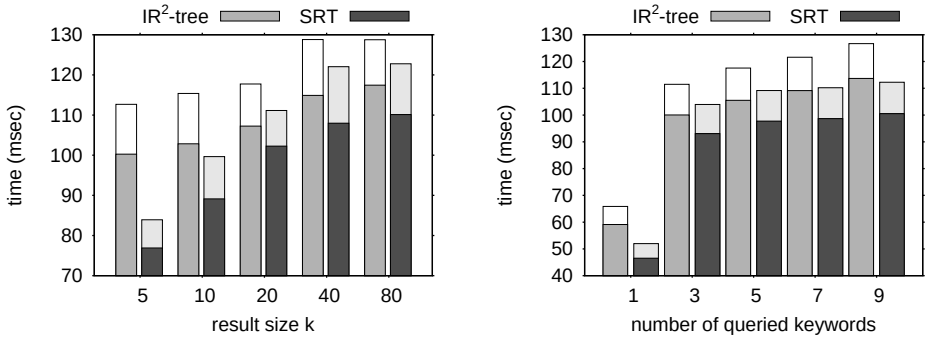
Figure 2.11 shows the execution time of *STPS* for the real dataset when varying query parameters. In Figure 2.11(a), time decreases for large k values compared to the range score (Figure 2.8(b)), because combinations with high score are associated with all data objects. Even though the score of the object is reduced based on the distance, still their score is high enough to retrieve fewer combinations. For smaller k values the execution time is not affected significantly. In

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) Varying k (b) Queried keywords

Figure 2.11: Influence query for real dataset.



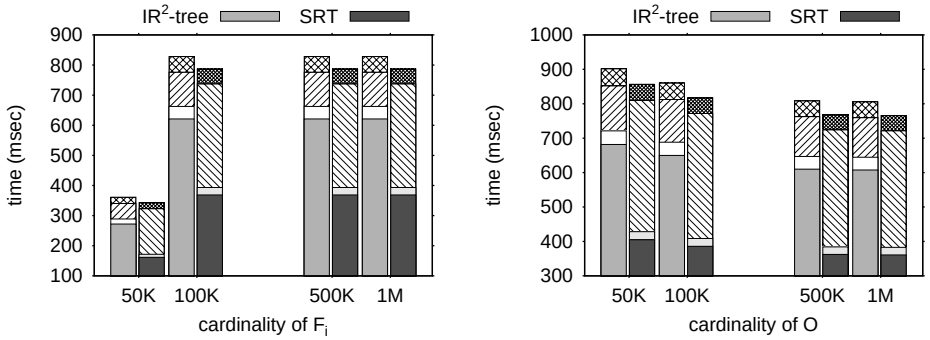
(a) Varying k (b) Queried keywords

Figure 2.12: Influence query for synthetic dataset.

Figure 2.11(b), we evaluate the performance of *STPS* when varying the number of queried keywords. We notice that the execution time is similar to Figure 2.8(d), which depicts the results of the same experiment for range score.

Finally, in Figure 2.12, we study the performance of *STPS* for the synthetic dataset when varying query parameters. The execution time is similar and slightly higher to the execution time needed for the range score (Figure 2.9), while the behavior of *STPS* when varying query parameters is the same. Again, the SRT indexing technique improves the performance of *STPS* consistently.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) Cardinality of F_i (b) Cardinality of O

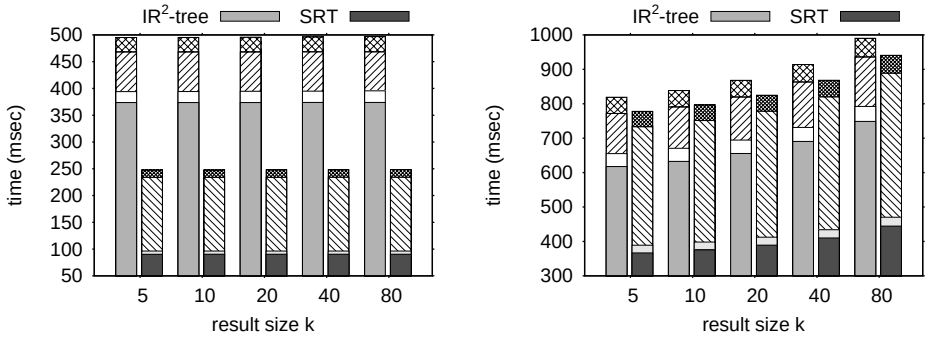
Figure 2.13: Scalability of nearest neighbor variant.

2.8.5 Nearest Neighbor Preference Score

In this section, we evaluate the performance of *STPS* for the nearest neighbor score variant. In general, we noticed that the execution time is higher compared to the other score variants, which is due to the Voronoi cell computations required for retrieving the data objects. In the charts, we illustrate separately with a striped pattern the IO (lower striped part) and the CPU-time (upper striped part) required to compute the respective Voronoi cells. Moreover, it is expected that for a given combination, few data objects satisfy the nearest neighbor constraint, which leads to retrieval of more combinations compared to the other variants. Therefore, we notice in the charts that the execution time is high even if the Voronoi cell computations is not considered (without stripped parts). We note that for static data the Voronoi cells can be pre-computed in a special structure, and therefore significantly reduce the execution time.

Figure 2.13 depicts the execution time for *STPS* for the synthetic dataset, while varying the size of the feature and object datasets. In Figure 2.13(a) we notice that for large feature sets the dominant cost is finding the data objects for a given combination (i.e., computing the Voronoi cells), rather than retrieving the combination with the highest score. Computing the Voronoi cells requires retrieval of feature objects from the spatio-textual index of F_i to define the borders of the cell. This cost is higher for the SRT indexing method compared to the IR^2-tree, since the IR^2-tree is built based on spatial information only and nearby feature objects are stored in the same node. Nevertheless, SRT indexing is still beneficial for *STPS*, but the gain is smaller than for the other variants. Similar conclusions can be drawn when varying the cardinality of the dataset O, as depicted in Figure 2.13(b).

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) Real dataset (b) Synthetic dataset

Figure 2.14: Varying k for nearest neighbor variant.

In Figure 2.14, we vary the parameter k both for real (Figure 2.14(a)) and synthetic datasets (Figure 2.14(b)). We notice that the execution time does not increase significantly when increasing k for the real dataset. This is because there exist some combinations for which their feature objects are the nearest neighbor for many data objects. Thus, the same effort is needed for retrieving few or many data objects. This is not the case for the synthetic dataset (Figure 2.14(b)), where the execution time increases for higher values of k.

2.9 Conclusions

Recently, the database research community has lavished attention on spatio-textual queries that retrieve the objects with the highest spatio-textual similarity to a given query. Differently, we address the problem of ranking data objects based on the facilities (feature objects) that are located in their vicinity. A spatio-textual preference score is defined for each feature object that takes into account a non-spatial score and the textual similarity to user-specified keywords, while the score of a data object is defined based on the scores of feature objects located in its neighborhood. Towards this end, we proposed a novel query type called *top-k spatio-textual preference query* and present two query processing algorithms. *Spatio-Textual Data Scan* (*STDS*) first retrieves a data object and then computes its score, whereas *Spatio-Textual Preference Search* (*STPS*) first retrieves highly ranked feature objects and then searches for data objects nearby those feature objects. Moreover, we proposed an indexing technique that improves the performance of our algorithms. Furthermore, we show how our algorithms can support

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

different score variants. Finally, in our experimental evaluation, we put all methods under scrutiny to verify the efficiency and the scalability of our method for processing top-k spatio-textual preference queries.

2.9.1 Acknowledgments

This research work was developed in cooperation with George Tsatsanifos and the research results where published at:
George Tsatsanifos, Akrivi Vlachou: On Processing Top-k Spatio-Textual Preference Queries, in Proceedings of 18th International Conference on Extending Database Technology (EDBT), Brussels, Belgium, March 23-27, 2015.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Chapter 3

Maximizing Influence Of Spatio-Textual Objects Through Keyword Selection

In modern applications, spatial objects are increasingly annotated with textual descriptions, thus offering the opportunity to users to formulate expressive *spatio-textual queries* that combine spatial distance with textual relevance. For example, given a database of hotels annotated with features (in the form of keywords) extracted from their web page, tourists can pose queries that retrieve a set of hotels ranked based on some combination of distance and textual similarity to the query keywords. In this context, a challenging problem is how to select a bounded set of at most b keywords to describe the facilities of a spatial object, in order to make the object appear in the top-k results of as many users as possible. We formulate this problem, called *Bests-terms*, using concepts related to top-k and reverse top-k queries, and show that it is NP-hard. Hence, we present a baseline algorithm that provides an approximate solution to the problem. Then, we introduce a novel algorithm for keyword selection that greatly improves the efficiency of query processing. By means of a thorough experimental evaluation using real data, we demonstrate the performance gains attained by our approach.

3.1 Introduction

Spatio-textual search has attracted increased attention recently, due to the numerous applications that provide value-added services to the users by combining spa-

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

tial location with textual relevance. Given a database of geographical points of interest that are annotated with textual information (also called *spatio-textual objects*), the objective of a spatio-textual query is to retrieve a ranked set of top-k spatio-textual objects that are close to the query point and have high textual similarity to the query keywords. As a notable example, consider hotels that are annotated with their facilities (e.g., in the form of keywords) and tourists that search for hotels close to some location of interest and a set of query keywords indicating desired facilities (for example "pool" or "Wi-Fi").

An interesting problem encountered in real-life applications that rely on spatio-textual retrieval is how to improve the ranking of a spatio-textual object for as many users as possible. For instance, for a newly established hotel at some location, the question is how to enrich its textual annotation in order to maximize its rank for many different users. To address this challenging problem, we capitalize on reverse top-k queries[50], which retrieve the set of users that have a given object in their top-k results. We model the problem as a maximization of the cardinality of the reverse top-k result set, and we explore the different combinations of keywords that will increase the query object's rank for many users, when added to its textual annotation. We call this problem as *Best terms*, we show that it is NP-hard, and we present a greedy solution that serves as baseline. Then, we propose a novel algorithm that boosts the performance of query processing, by deliberately selecting keywords that increase the score of the query object for many users simultaneously. Finally, we present the results of our experimental evaluation that verifies the performance gains of our algorithm.

In summary, our main contributions are outlined below:

- We formulate the novel problem, called *Best terms*, of increasing the rank of a spatio-textual object for many different users, by enriching its textual description.
- We show that the *Best terms* problem is NP-hard and we provide a baseline solution.
- We propose an efficient query processing algorithm that significantly outperforms the baseline consistently.
- We provide an experimental evaluation that demonstrates the merits of our approach.

The rest of this chapter is structured as follows: Section 3.2 provides an overview of the related work. Section 3.3 presents the necessary background and

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

preliminary concepts. Then, in Section 3.4, we formally describe the problem statement. Section 3.5 presents the baseline algorithm, while Section 3.6 describes our efficient query processing algorithm. Section 4.4 shows the experimental evaluation, and Section 3.8 concludes the chapter.

3.2 Related Work

In this section, we provide an overview of the related research literature.

Keyword recommendation. Zhang et al. [56] present a method for recommending keywords for advertisements in keyword search results using Wikipedia. They focus mostly in cases where the advertisement (target) consists of short-text web pages which contain inadequate textual content to describe the advertised entity. Based on the fact that a large number of entities are described in Wikipedia, they use Wikipedia articles relevant to the advertised entity in order to recommend keywords to connect to the target. Fuxman et al. [19] follow a different approach. They suggest keyword queries to advertisers using logs which store the queries posed by the users and the URLs of the result set that were selected by the users. Some of the URLs are also connected to a set of concepts. The target of the authors is to connect the set of concepts to the queries using the Markov Random Field model and suggest the most relevant queries for each concept to the advertisers. Ravi et al. [36] propose variety of methods for automatic generation of bid phrases. Among others they introduce the usage of a translation model that extends a predefined mapping between bidding phrased and target web pages. Papadimitriou et al. [35] study the problem of mapping an advertisement in a set of URLs based on keyword queries. In particular they assume that each advertisement is mapped to a set of keyword queries and their aim is to map each advertisement in a set of URLs which will be representative of the results produced by the attached keyword queries. Choi et al. [12] create a representative summary of the advertisement based on the context of the advertised material. Their method is making use of co-occurrence and semantic vectors in order to enrich the ad context and create a representative set of terms. Cholette et al. [13] study the problem of finding optimal bids in search based algorithms. Agrawal et al. [2] introduce an approach for recommending bid phrases from a given ad landing page by classifying a set of labels generated by click logs. Their classifier has logarithmic complexity and can efficiently make predictions on large sets of labels.

The aim of the aforementioned approaches is to identify potentially relevant

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

queries to the advertised products and form bid phrases based on the identified queries. Our approach is inherently different because the above techniques try to predict relevant queries and do not consider the relevance of the advertised product in relation to similar products. In addition they do not consider top-k search criteria as the appearance of a product in a search result is decided mainly on the bidding strategy. On the contrary our aim is to enhance the description of a spatio-textual object and to increase the number of queries for which the target product appears in the top-k list of the search results. In this effort we take into consideration not only the user preferences but also the rest of the spatio-textual objects which are relevant to those queries.

Spatial Keyword Search. Spatial keyword search has been well studied during the recent years and several index structures have been introduced for efficient search. A detailed evaluation of existing spatio-textual indexes can be found in [11]. Felipe et al. [17] introduced the IR^2-tree index which integrates a bitmap signature on each node of an R-tree describing the textual content of the subtree rooted at the node. Cong et al. [14] introduced the IR-tree and its variants. The IR-tree is based on the R-tree structure as well. Each node of the tree is also associated with inverted index containing the textual information of the children of the node. Rocha et al. [38] proposed the S2I index which uses different strategies for frequent and infrequent terms. The spatial distribution of a frequent term is stored in an aggregated R-tree (aR-tree) where each node contains an aggregated value of the impact of the term on the objects contained in the subtree rooted at the node. Cao et al. [7] introduce the concept of *prestige* where a spatio-textual object has a higher prestige if it is collocated with other textually similar objects. They calculate the prestige of a spatio-textual object based on a graph where each node corresponds to an object and two nodes are connected if and only if their textual similarity and spatial proximity exceed certain thresholds.

Ying Lu et al. [32] and Jiaheng Lu et al. [31] studied the problem of reverse spatial and textual k nearest neighbor search, where, given a query point q, the objective is to locate the set of spatio-textual objects for which q is among the k nearest neighbors. The distance between the objects is a linear combination of the textual and the euclidean distance of the objects. The authors introduce the IUR-tree which is an adaptation of the IR-tree. Each node of the IUR-tree contains the union and the intersection of the terms contained in the objects in the subtree rooted at the node. Our approach is different as we do not evaluate the similarity between elements of a set of spatio-textual objects, but our aim to increase the relevance and therefore the visibility of an object against a set of user preferences which constitutes a different set from that of the spatio-textual objects that our

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

query object belongs.

Wu et al. [51] propose the W-IR-tree which is similar to the IR-tree but it differs in the way it is constructed. While the IR-tree places the objects in leaf nodes based on their distance, the W-IR-tree partitions the objects based primarily on their textual relevance. The W-IR-tree shows improved performance for batch queries where objects are considered relevant to the query only if they contain all terms of the query. The W-IR-tree cannot be applied in our case as we consider it possible for a spatio-textual object to be relevant to the a user preference even if it does not contain all terms of the user preference.

3.3 Preliminaries

Let D be a set of objects where each object o is represented by a tuple of the form $o = \langle o.T, o.L \rangle$ where $o.T$ is a set of keywords describing the features of o and L is a point in $\mathbb{R}^2$ describing the location of o. We denote as $\mathcal{A} = \bigcup_{o \in D} o.T$ to be the set of all keywords in D. In the scope of this chapter we call these objects *spatio-textual objects*. For a given object o, we consider the *size* of o to be equal to $|o.T|$, namely the size of an object is the number of terms it contains.

3.3.1 Top-k spatial keyword queries

Let u be a user preference query on D, where u is represented by the a tuple $u = \langle u.T, u.L, \alpha \rangle$, $u.T \subseteq \mathcal{A}$ is the text describing the user's desired features, $u.L \in \mathbb{R}^2$ denotes the desired location and $\alpha \in \mathbb{R}$ denotes the importance of location over matching the desired features. Given a preference u, we can assign a score to each object using the following equation:

$$f(o, u) = \alpha \times \delta(o.L, u.L) + (1 - \alpha) \times \theta(o.T, u.T) \tag{3.1}$$

where $\delta(o.L, u.L)$ is the spatial distance, and $\theta(o.T, u.T)$ is the textual distance between the object o and the user preference u. Given an integer k, we can return the top-k spatio-textual objects according to their score. In the scope of this chapter, we assume that lower scores are better, both spatial and textual distances are normalized in the interval $[0, 1]$ and $f(o, u) = 1.0$ if $\theta(o.T, u.T) = 1$. The latter assumption implies that objects that are not textually relevant to the query cannot be considered as a valid result.

The textual relevance we employ is the normalized intersection of terms between the description of a spatio-textual object $o.T$ and a user preference keyword

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

set $u.T$, i.e., $\theta(o.T, u.T) = 1 - |o.T \bigcap u.T||u.T|^{-1}$. Although in large documents different textual similarity functions are more appropriate, the intersection is more representative in cases of feature selection. For instance if a user is looking for a hotel with a restaurant and a pool, any hotel offering more features (e.g. restaurant, pool, bar) than the ones specified by the user should not be less textually relevant than a hotel which offers only the features specified by the user preference (restaurant, pool).

Definition 1 ***Top-k query.*** *Given a set D of spatio-textual objects, a set of terms $\mathcal{A}$, a scoring function f, an integer k, and a query u, the result set $TOP_k(u)$ of a top-k query is a set of spatio-textual objects such that $TOP_k(u) \subseteq D$, $|TOP_k(u)| = k$ and $\forall o_1, o_2 : o_1 \in TOP_k(u), o_2 \in D - TOP_k(u)$ it holds that $o_1.T \bigcap u.T \neq \emptyset$ and $f(o_1, u) \leq f(o_2, u)$.*

If an object o belongs to the $TOP_k(u)$ set of a user preference u, we say that o is *visible* to u or that u *sees* o. For a specific set of objects D and a set of user preferences U, it is possible to identify for a query object q the set of users who can see q. This is the reverse procedure of a top-k query and therefore it is called *reverse top-k ($RTOP_k$) query* [46].

Definition 2 ***$RTOP_k$ query.*** *Given a set D of spatio-textual objects, a set of user queries U, a scoring function f, integer k, and a spatio-textual object q, the result set $RTOP_k(q)$ of a reverse top-k query is set such that $RTOP_k(q) \subseteq U$ and $u \in RTOP_k(q)$ if and only if $\exists o \in TOP_k(u)$ such that $f(q, u) \leq f(o, u)$.*

The cardinality of the $RTOP_k$ set of a query-object q is called *influence score* of the object and we denote it as $I(q)$. The influence score indicates the number of users to whom q is visible.

3.3.2 IR-tree

We employ a state-of-the-art index structure to process spatial keyword queries, namely the IR-tree [14]. The IR-tree is an R-tree where each node is associated with an inverted index of the objects contained in the respective sub-tree rooted at the node. In more detail, each leaf node contains an inverted index of the spatio-textual objects contained in the node. The leaf node is characterized by a spatio-textual pseudo-object which consists of a *minimum bounding rectangle* (MBR) which encloses all objects of the node and a pseudo-document which consists

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

of the union of all the terms contained in the children of the node. Each non-leaf node contains an inverted index of the spatio-textual pseudo-objects of the children nodes it contains. Non-leaf nodes are also characterized by spatio-textual pseudo-objects which are constructed similarly to the pseudo-objects of the leaf nodes.

3.4 Problem Definition

As mentioned earlier, given a set of spatio-textual objects D and a set of spatio-textual preferences U, the influence score of an object q is the number of preferences to which q is visible. Assuming that the location of a spatio-textual object cannot change, the only the way to improve the influence score of q is to enhance its textual description, in order to increase the textual relevance between q and the user preferences in U. We study the problem finding a set of b terms which when added to the textual description of q, they maximize the influence score q. We refer to this problem as *Best-terms query*.

Definition 3 ***Best-terms query.*** *Given a set D of spatio-textual objects, a set of terms $\mathcal{A} = \bigcup_{o \in D} o.T$, a set of queries U, a scoring function f, an integer k, a spatio-textual object $q = \langle q.T, q.L \rangle$, and an integer b, the set BT is a set of terms such that $\mathrm{BT} \subseteq \mathcal{A}$, $\mathrm{BT} \bigcap q.T = \emptyset$, $|\mathrm{BT}| \leq b$ and $\forall T \subseteq \mathcal{A}, |T| \leq b$ it holds that $I(q_1) \geq I(q_2)$ where $q_1 = \langle q.T \bigcup \mathrm{BT}, q.L \rangle$ and $q_2 = \langle q.T \bigcup T, q.L \rangle$.*

The Best-terms problem is NP-hard. We show that by studying a special case of a Best-terms query, namely the respective decision problem of finding whether there exists a set of terms T with $|T| \leq b$ such that $I(\langle q.T \bigcup T, q.L \rangle) = |U|$.

Definition 4 ***Best-terms query (decision problem).*** *Given a set D of spatio-textual objects, a set of terms $\mathcal{A} = \bigcup_{o \in D} o.T$, a set of queries U, a scoring function f, an integer k, and a spatio-textual object $q = \langle q.T, q.L \rangle \in D$, decide if there is a set BT such that $\mathrm{BT} \subseteq \mathcal{A}$, $\mathrm{BT} \bigcap q.T = \emptyset$, $|\mathrm{BT}| \leq b$ for which it holds that $I(q_1) = U$ where $q_1 = \langle q.T \bigcup \mathrm{BT}, q.L \rangle$*

We will show that Problem 4 is NP-complete by reducing the set cover problem in Problem 4 using the restriction technique [20].

Definition 5 ***Set cover problem.*** *Let U be a set of elements (universe) and $\mathcal{T} = \{T_1, \ldots, T_n\}$ be a collection of sets where $\bigcup_{i=1}^{n} T_i = U$. The* set cover problem *decides if there is a subset of $\mathcal{T}$, $\mathcal{T}' \subseteq \mathcal{T}$ of size $|\mathcal{T}| \leq b$ such that $\mathcal{T}'$ is a cover of U.*

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Proof. Let an oracle machine select the BT set for a query object q. We set $p = \langle q.T \bigcup \mathrm{BT}, q.L \rangle$ and by performing a TOP_k query for each user preference we can calculate the $RTOP_k(p)$ set and the influence score $I(p)$ of object p in polynomial time. Therefore the solution can be verified in polynomial time and our problem belongs to the NP class.

We set U a to be a set of users and $D = \{q\}$. We define a collection $\mathcal{T} = \{T_1, \ldots, T_{|A|}\}$ of sets, one for each term t_i in A where a user u belongs in T_i only if $t_i \in u.T$. If we consider $k = 1$, then, for all users that $q.T \bigcap u.T = \emptyset$ it holds that $q \notin TOP_k(u)$ since q is not relevant to $u.T$. If $q.T \bigcap u.T \neq \emptyset$ then $q \in TOP_k(u)$ as it is the only object. Therefore any selection of a term t_i is equivalent of selecting a subset of T_i of U. The set cover problem can therefore be seen as a special case of Problem 4 and therefore Problem 4 is NP-complete which leads us to the conclusion that the Best-terms problem is NP-hard.

3.5 Baseline

In this section we describe a baseline approximate solution for the Best-terms problem. An exact solution is infeasible to be calculated as the the problem is NP-hard. Therefore we use a greedy algorithm, HRJN (Best Term First), which on each step adds to the query object the term that induces the highest increase in influence score.

Algorithm 6 describes the HRJN approach. HRJN takes as input an IR-tree index containing the set of spatio-textual objects D, and an IR-tree index containing the set of user preferences U. It starts by creating a pseudo-preference q', in order to traverse the preferences based on their distance to the query object q. It first creates a set C of candidate spatio-textual objects, one for each term that can be added to q. The size of C is equal to $|\mathcal{A} - q.T|$. In lines 10,11 the algorithm checks if performing a top-k query is necessary. It calculates the score of the last retrieved spatio-textual objects with the scores of the candidate objects and if there are k objects which have a better score than all objects in C, the user preference is ignored as no candidate object can be in the TOP_k set of the current user preference. In the opposite case the top-k query is executed and the TOP_k result set is stored in the buffer. All candidate objects which are no worse than the k-best element of the calculated TOP_k set belong also to the TOP_k set of u and therefore their influence score is increased. When all user preferences have been examined, the object with the highest influence score is selected and a new

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Algorithm 6 HRJN

Input: U:set of users,D: set of objects,
q:query point, b : number of new terms
Output: BT: set of new terms

1: $C \leftarrow \emptyset$,
2: $q' \leftarrow \langle q.T, q.L, 1\rangle$
3: bestCandidate$\leftarrow q$
4: **for** $i = 0; i < b; i++$ **do** *//repeat until b new terms have been found*
5: **for all** $t \in \mathcal{A} - o.T$ **do**
6: $C \leftarrow C \bigcup \{\langle \text{bestCandidate}.T \bigcup \{t\}, \text{bestCandidate}.L\rangle\}$
7: **end for**
8: $u \leftarrow$next(U,q')
9: **while** $u \neq null$ **do**
10: $\tau \leftarrow \max_{p \in \text{buffer}} (f(p,u))$
11: **if** $\exists c \in C : f(c,u) \leq \tau$ **then**
12: buffer$\leftarrow \text{TOP}_k(u)$
13: $\tau \leftarrow \max_{p \in \text{buffer}} (f(p,u))$
14: **for all** $c \in C$ **do**
15: **if** $f(c,u) \leq \tau$ **then**
16: $I(c) \leftarrow I(c) + 1$
17: **end if**
18: **end for**
19: $u \leftarrow$next(U,q')
20: **end if**
21: **end while**
22: bestCandidate$\leftarrow \text{argmax}_c(I(c))$
23: **end for**
24: BT$\leftarrow$ bestCandidate.T-q.T
25: **return** BT

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

set of candidate objects is created based on that object. The procedure is repeated b times until an object with b new terms is created. The terms that are included in the resulting object and not in q constitute the resulting BT set.

The IR-tree index on U helps us reduce the number TOP_k queries executed. As the preferences are accessed by ascending distance, the score of q is expected to reduce and therefore it becomes less likely for a TOP_k query to be executed.

3.6 Graph Based Term Selection

HRJN extends the textual description of a spatio-textual object incrementally, fact that forces the algorithm to scan the preferences multiple times leading to a high computational cost. In the section we present a new algorithm GBTS (Graph Based Term Selection) which examines the set of preferences only once per query object and creates a graph of terms which provides an estimation of the influence gain any combination of terms may provide.

GBTS consists of two separate algorithms. The first algorithm GC (Graph Construction) creates a graph connecting the terms which when added to a spatio-textual object q, they can induce an increase in its influence score. The second algorithm BSS (Best Subgraph Selection) traverses the graph identifying the sets of terms which will induce the highest increase in the influence score of q. In more detail, given a set of objects D, a set of user preferences U and a spatio-textual object q, we denote as $\widehat{U}(q)$ the set of all preferences for which q is not visible and at most b terms are needed for q to become visible. The first algorithm, GC, constructs a weighted graph $G = (V, E)$ where each node of the graph represents a candidate term, and the edges connecting the nodes indicate the maximum increase in the influence score of q if a set of terms is added to q. For each examined user preference u, the algorithm adds to the graph a node for each previously unseen term. If one additional term is enough for u to be added to $RTOP_k(q)$, the algorithm adds a loop edge with weight equal to 1, to each term t that is not contained in q. If the edge already exists, the weight is simply added to the existing edge. In cases where more than one terms are necessary for q to be included to $TOP_k(u)$, the procedure is slightly different. Let $T = u.T - q.T = \{t_1, \ldots, t_n\}$ be the terms that are included in u but not in q and $1 < \lambda \leq n$, be the number of terms that need to be added to u for it to be included in the $RTOP_k(q)$. For each pair of terms in $u.T - q.T$, the algorithm adds an edge with weight equal to $2\,(\lambda(\lambda-1))^{-1}$. The sum of weights of the edges added to each subgraph $G' = (V', E')$ where $V' \subseteq T$ and $|V'| = \lambda$ is equal to 1 indicating the potential

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Algorithm 7 GC

Input: U:set of users,D: set of objects,
q:query point, b : number of new terms
Output: $G = (V, E)$: resulting graph

1: $V = \emptyset, E = \emptyset, G = (V, E)$ *//graph initialization*
2: $q' \leftarrow \langle q.T, q.L, 1\rangle$
3: $u \leftarrow next(U, q')$
4: **while** $u \neq null$ **do**
5: buffer$\leftarrow \mathrm{TOP}_k(u)$
6: $\tau \leftarrow \max_{p \in \text{buffer}} (f(p, u))$
7: **if** $f(q, u) > \tau$ **then** *//if $q \notin TOP_k(u)$*
8: $T \leftarrow u.T - q.T$
9: $V \leftarrow V \bigcup T$
10: $\lambda \leftarrow \max\left(1, \left\lceil \left(1 - \frac{\tau - a\delta(q,u)}{1-a}\right)|u| \right\rceil\right)$
11: **if** $\lambda \leq 1$ **then**
12: $E \leftarrow E \bigcup \{e = (t_i, t_i, 1) : t_i \in T\}$
13: **else if** $1 < \lambda \leq b$ **then**
14: $E \leftarrow E \bigcup \left\{e = \left(t_i, t_j, \frac{2}{\lambda(\lambda-1)}\right) : t_i, t_j \in T \text{ and } t_i \neq t_j\right\}$
15: **end if**
16: **end if**
17: $u \leftarrow$next(U,q')
18: **end while**
19: **return** G

increase in the influence score of q if the terms contained in G' are added to q. As before, if an edge already exists, the weight is added to the existing edge.

When the graph has been created, algorithm BSS (Best Subgraph Selection) chooses as seed nodes the b nodes (terms) of the graph with the highest degree and creates a set of b subgraphs with initially one node each. Next, each subgraph is expanded by adding at each step the node with highest degree that is adjacent to the subgraph. The expansion of each subgraph is continued until each subgraph has b nodes or the subgraph cannot be expanded. Finally the subgraph with the highest sum of edges is selected as solution and the set of terms included in the subgraph are the ones that constitute the BT set.

Algorithm 7 describes the construction of the term graph G. GC starts with

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Algorithm 8 BSS

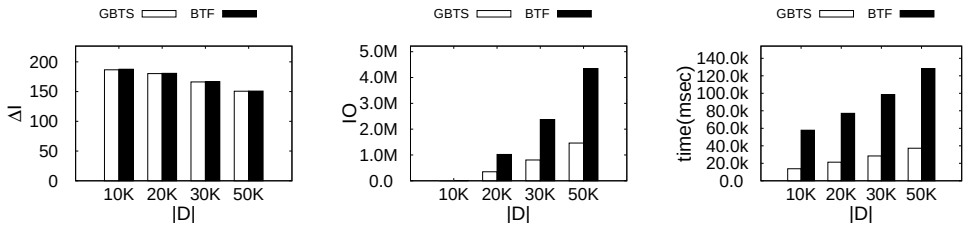
Input: $G = (V, E)$: graph, b: number of desired terms
Output: BT:set of new terms

1: $Q \leftarrow \emptyset$ *//Priority Queue*
2: BT $\leftarrow \emptyset$
3: **for** $i = 0; i < b; i++$ **do**
4: $t_i \leftarrow$ next node of G with the highest degree
5: $G_{t_i} \leftarrow$ expandNode(t_i)
6: Q.add(sumOfWeights(G_{t_i}),G_{t_i})
7: **end for**
8: **while** $|\text{BT}| \leq b$ **do**
9: $G_S \leftarrow$Q.pop()
10: add to BT the $b - |\text{BT}|$ highest degree nodes from G_S
11: **end while**
12: **return** BT

creating a pseudo-preference in order to traverse the preferences based on their distance to q. For each user preference if q is not in the $TOP_k(u)$ set, GC updates the node set of G and calculates λ, the number of terms that need to be added in q for it to be included in the $TOP_k(u)$ set. The number of terms is calculated based Equation 3.1 and the minimum score q is required to have in order to be in the $TOP_k(u)$ set. A non-positive value of λ indicates that u is located near q but $q.T \bigcap u.T = \emptyset$ and therefore q is not included in the $TOP_k(u)$ set. The addition of any term will allow q to be added to $\text{TOP}_k(u)$ set and therefore one loop edge is added to each term t for which it holds $t \in u.T - q.T$. If more than one terms are necessary to be added in q ($\lambda > 1$), GC adds all necessary edges in the graph. The algorithm continues until all user preferences have been examined. The size of the graph depends on the number of distinct terms contained in $\widehat{U}(q)$. The terms correspond to the features extracted from the textual descriptions of spatio-textual objects that describe the offered facilities. In practice, we have noticed that the vocabulary for the targeted applications is limited and therefore the graph is expected to fit in the main memory.

Algorithm 8 describes the algorithm of term selection. Initially an empty priority queue (Q) is constructed. Subsequently, at line 4 the algorithm chooses as seed the highest degree node t_i that has not yet been selected and constructs the subgraph G_{t_i} (line 5). The subgraph is constructed by repeatedly selecting the highest degree node adjacent to the G_{t_i} until $|G_{t_i}| = b$ or until no nodes can be

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) ΔI (b) I/O (c) time

Figure 3.1: Varying data cardinality

added to G_{t_i}. When each subgraph is constructed, it is pushed to Q. The sorting key of Q is the sum of weights of the edges in the subgraph. The BT set is constructed by selecting the subgraph with the highest sum of edges and adding the terms of the subgraph to BT. If the subgraphs contain less than b terms, more subgraphs are pulled from the priority queue until BT contains b terms. In such cases we add from each subsequent subgraph to BT the $b - |\mathrm{BT}|$ highest degree nodes of the subgraph.

3.7 Experimental Evaluation

In this section, we present the results of the experimental evaluation. All algorithms were implemented in Java and the experiments were executed on an AMD Opteron 4130 Processor (2.00GHz), with 32GB of RAM and 2TB of disk.

Datasets and metrics. For the data set D of spatio-textual objects, we used a set of 200000 descriptions of hotels from the site of Booking.com[1]. The dataset contains 190 distinct features. The set of preferences U was generated using a uniform distribution for creating the location of each preference while the terms were randomly chosen from the vocabulary generated by processing the set of hotels. The location of the user preferences was bounded in the MBR defined by set of hotels. We also tested our algorithm against a Zipfian distribution of terms. We used the Zipfian distribution generator provided by the Apache Commons project[2]. The metrics under which we evaluated the implemented algorithms were: a) increase in the influence score ΔI, b) number of I/O's performed by each algorithm, and c) processing time.

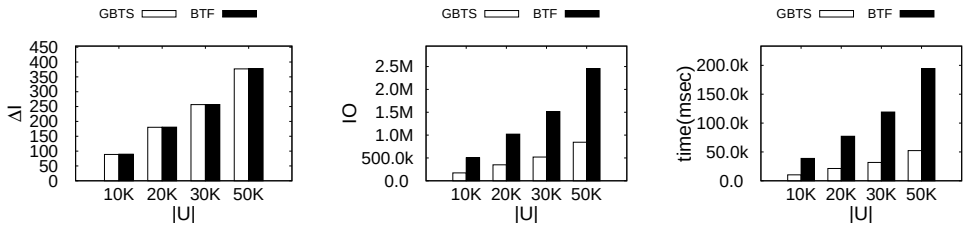
Experimental procedure. Both datasets D and U were indexed using an IR-

[1]http://www.booking.com

[2]http://commons.apache.org/proper/commons-math/

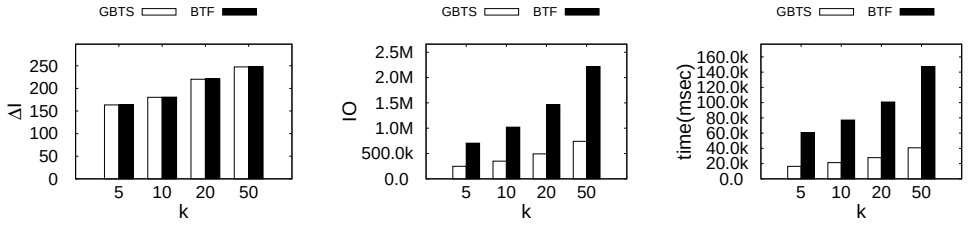
The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union



(a) ΔI (b) I/O (c) time

Figure 3.2: Varying preferences cardinality



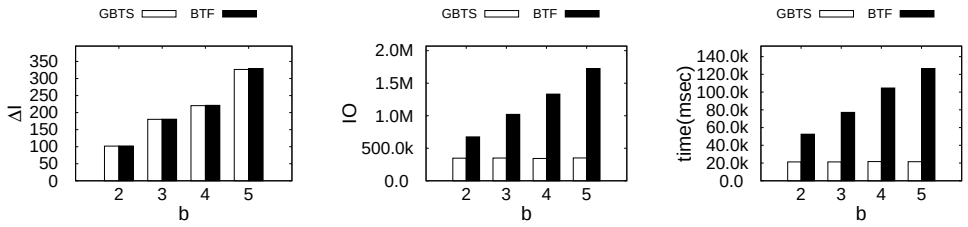
(a) ΔI (b) I/O (c) time

Figure 3.3: Varying k

tree where the maximum capacity of each node was 100 entries. We employed a buffer which was fixed at the size of 4MB, both for the tree index and for the inverted files. We run a series of experiments varying the parameters of a) the cardinality of D in the interval [10K,200K], b) the cardinality of U, [10K,200K], c) the number of returned results per user preference k, [5,50], d) the maximum size of user preferences, [1,5], and d) the number of returned terms for a query object b, [2-5]. For the Zipfian distribution we varied the value of the characteristic exponent s in the interval [0.1-1.0]. The default setup for the experiments was: $|D| = 20K$, $|U| = 20K$, $k = 10$, $b = 3$ and each the maximum preference size was set to 5. For each experiment a random set of 20 query objects was selected from D.

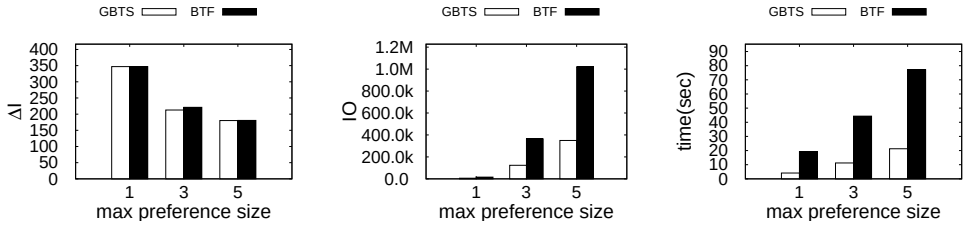
Varying $|D|$. Figure 3.1 illustrates the performance of the algorithms as we vary the number of spatio-textual objects. Figure 3.1(a) indicates that both algorithms perform similarly with respect to the increase of the influence score. As the number of objects increase the gain in influence score drops as more spatio-textual objects compete for the same number of user-preferences and therefore it becomes harder for a query object to increase its influence score. Figures 3.1(b) and 3.1(c) indicate that the I/O accesses and the processing time for both algorithms increase when the dataset size raises. As the dataset size increases the cost of a single

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) ΔI (b) I/O (c) time

Figure 3.4: Varying b



(a) ΔI (b) I/O (c) time

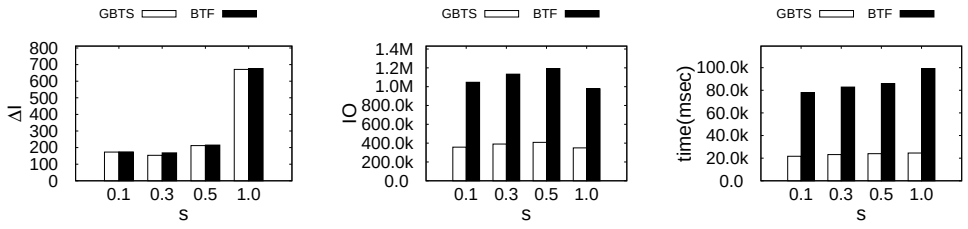
Figure 3.5: Varying max preference size

TOP_k query increases as well and therefore both algorithms are affected by the dataset size. The effect on HRJN is larger than in GBTS as HRJN accesses the data multiple times in order to create the set of new terms.

Varying $|U|$. Figure 3.2 depicts the performance of both algorithms as more preferences are processed. When the number of preferences increases there are more user preferences that can be added to the RTOP_k set of an object with an addition of a new set of terms and therefore the gain in influence score increases as well. The processing cost for both algorithms is expected to raise for a larger number of user preferences, as more preferences have be to examined. Both processing time and I/O cost raise faster for HRJN than for GBTS. In particular the processing cost for HRJN grows almost by a factor of b faster than GBTS as HRJN has to process the set of preferences b times in order to identify the set of new terms.

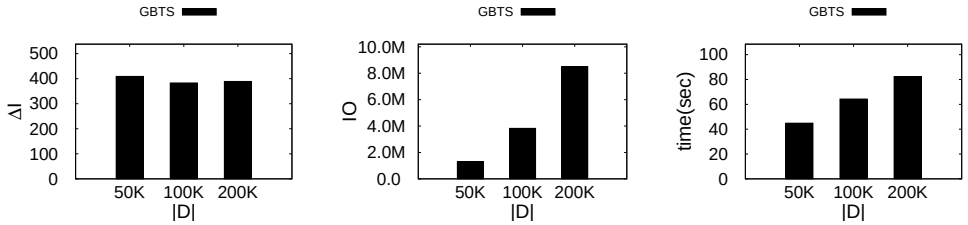
Varying k. As the size of the TOP_k set of each preference increases, the cost of a single TOP_k query increases as well. Figure 3.3 indicates that the increased I/O and processing cost of a TOP_k query affects both algorithms but similarly to increase on the size of datasets the effect on HRJN is magnified by a factor of b. The influence score gain raises as well, since with the increase in k more objects can be included in the TOP_k set of a user preference and the necessary increase

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) ΔI (b) I/O (c) time

Figure 3.6: Varying zipf distribution



(a) ΔI (b) I/O (c) time

Figure 3.7: Varying data cardinality

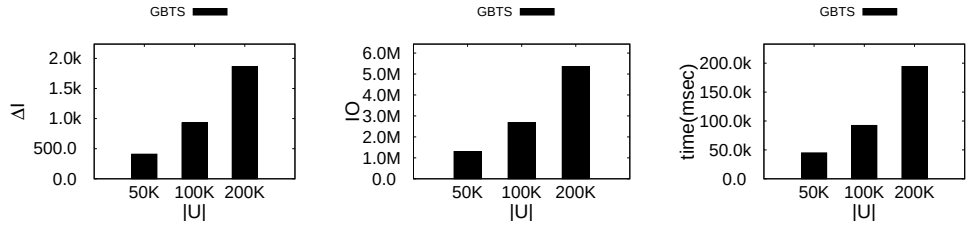
in the text similarity for a query object q to be added to a TOP_k set of a user preference u becomes smaller.

Varying b. Figure 3.4 illustrates the performance of the algorithms as we vary the number of new terms added to each query object. It is noteworthy the fact that both algorithms behave similarly with respect to the increase of the influence score. The cost of HRJN raises linearly with respect to b, which is expected as it has to process the data b times before returning the resulting BT set. On the other hand, GBTS remains unaffected by the increase of the b parameter as it has to access the preferences set only once.

Varying the query size. Figure 3.5 indicates that as the maximum preference size increases, the possible gain of influence score for a spatio-textual object drops. The reason lies in the fact that for a large user preference u, more terms are required to be added to a spatio-textual object q, for q to enter the $TOP_k(u)$ set. Larger queries require more complex TOP_k queries on the indexes and consequently the performance of both algorithms is affected. As expected HRJN is affected in a larger degree than GBTS by the increased cost of the TOP_k queries.

Zipfian distribution. It is quite usual the terms of user-preferences to follow a Zipfian distribution. We tested our algorithms against a set of user preferences where the occurrences of terms follow a Zipfian distribution. Figure 3.6 illustrates

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) ΔI (b) I/O (c) time

Figure 3.8: Varying data cardinality

the experimental results. Similarly to the uniform distribution, GBTS outperforms HRJN in terms of I/O accesses and processing time while producing the same gain in influence score. In cases where the exponent of the Zipfian distribution takes high values the gain in influence score raises significantly. Such behavior is expected as when a small number of distinct terms appear in a large number of user preferences, adding those terms to a spatio-textual object will result in a significant increase of its influence score since the addition of those terms will allow it to enter the TOP_k set of many user preferences.

Scalability analysis. We evaluated the performance of GBTS against larger datasets to evaluate the scalability of our approach. HRJN is not included in the results as it needed excessive time to produce results. The experimental results shown in Figures 3.7 indicate that the processing time of GBTS grows logarithmically with respect to the size of the dataset while the I/O cost increases linearly. In the first TOP_k queries we have an increased number of I/Os, however after a certain number of queries, several nodes of the IR-tree are buffered and as a result the subsequent TOP_k queries induce a limited number of I/O accesses. Figure 3.8 illustrates the performance of GBTS with respect to the cardinality of user preferences set. Both the processing time and the I/O increase linearly with respect to time.

3.8 Conclusions

In this chapter, we address the challenging problem of increasing the influence of a spatio-textual object, by enriching its textual description with at most b carefully selected keywords. In this way, the spatio-textual object's textual relevance to user queries is increased, with the ultimate objective being for the object to become part of the top-k result for many different users. We provide a formal problem

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

statement that is novel and relies on concepts related to top-k and reverse top-k queries. We show that the problem is NP-hard, and we present a greedy solution to the problem. Then, we propose a more efficient algorithm that achieves results of comparable quality, but with significantly lower processing cost. We demonstrate the performance gains of the proposed approach by means of a thorough experimental evaluation that includes real data.

3.8.1 Acknowledgments

This research work was developed in cooperation with Orestis Gkorgkas, Christos Doulkeridis and Kjetil Nørvåg and the research results are under submission:
Orestis Gkorgkas, Akrivi Vlachou, Christos Doulkeridis, Kjetil Nørvåg: Maximizing Influence of Spatio-Textual Objects through Keyword Selection submitted for publication.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Chapter 4

Location-aware Tag Recommendations for Flickr

Flickr is one of the largest online image collections, where shared photos are typically annotated with tags. The tagging process bridges the gap between visual content and keyword search by providing a meaningful textual description of the tagged object. However, the task of tagging is cumbersome, therefore tag recommendation is commonly used to suggest relevant tags to the user and enrich the semantic description of the photo. Apart from textual tagging based on keywords, an increasing trend of geotagging has been recently observed, as witnessed by the increased number of geotagged photos in Flickr. Geotagging refers to attaching location-specific information to photos, namely about the location where a photo was captured. Even though there exist different methods for tag recommendation of photos, the gain of using spatial and textual information in order to recommend more meaningful tags to users has not been studied yet. We analyze the properties of geotagged photos of Flickr, and propose novel location-aware tag recommendation methods. For evaluation purposes, we have implemented a prototype system and exploit it to present examples that demonstrate the effectiveness of our proposed methods.

4.1 Introduction

Flickr allows users to upload photos, annotate the photos with tags,view photos uploaded by other users, comment on photos, create special interest groups etc. Currently, Flickr stores one of the largest online image collections with more than

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

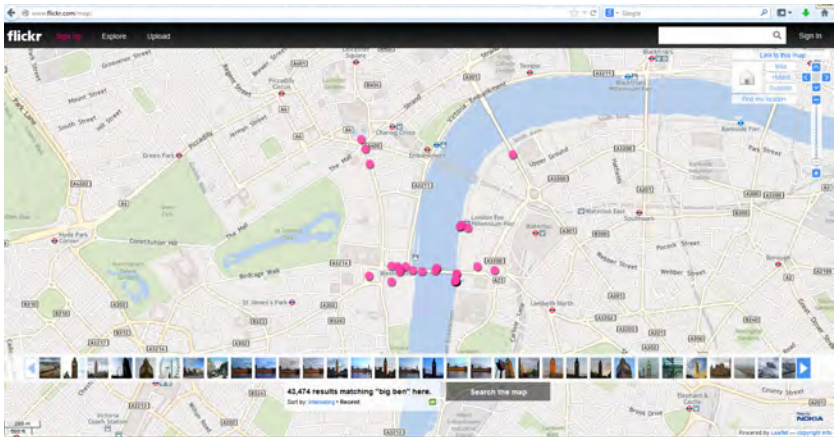


Figure 4.1: Example of geotagged photos on a map in Flickr.

8 billion photos (March 2013[1]) from more than 87 million users and more than 3.5 million new images uploaded daily. The tags are important for users to retrieve relevant photos among the huge amount of existing photos. Since multimedia data provide no textual information about their content, tags bridge the gap between visual content and keyword search by providing a meaningful description of the object. Thus, to make their photos searchable, users are willing to annotate their uploaded images with tags [3]. Nevertheless, tags reflect the perspective of the user that annotates the photo and therefore different users may use different tags for the same photo. This can be verified by the fact that photos of Flickr that depict the same subject may be described by a variety of tags. Tag recommendation [42] is commonly used to provide to the user relevant tags and enrich the semantic description of the photo.

Flickr motivates its users to geotag their uploaded photos[2]. Geotagging means to attach to a photo the location where it was taken. Photos taken by GPS-enabled cameras and mobile phones are geotagged automatically and location metadata, such as latitude and longitude, are automatically associated with the photos. Flickr is able to read the spatial information (latitude and longitude) during the upload and place the photos on a map, as depicted in Figure 4.1. Furthermore, photos may be also geotagged manually by the user when the photo is uploaded. Currently, there is an increasing trend in the number of geotagged photos in Flickr.

[1]http://www.theverge.com/2013/3/20/4121574/flickr-chief-markus-spiering-talks-photos-and-marissa-mayer

[2]http://www.flickr.com/groups/geotagging/

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Even though several recent studies [14, 6] examine how relevant web objects can be retrieved based on both the spatial and textual information, the gain of using spatial information in order to recommend more meaningful tags to users has not been studied yet. Nevertheless, it is expected that nearby photos may depict similar objects, thus sharing common tags with higher probability. In this chapter, we propose methods for tag recommendations based on both location and tag co-occurrence of the photos. In details, this work makes the following contributions:

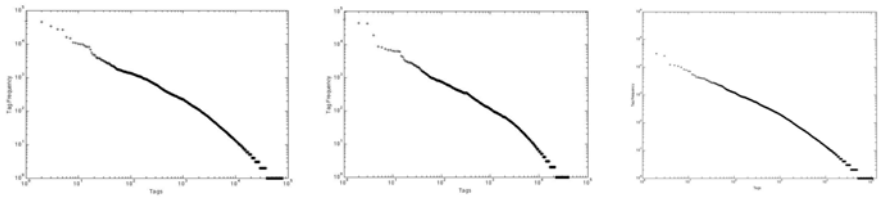
- We create different data collections of geo-tagged photos of Flickr that are located in different cities and analyze their properties in terms of tag frequency, number of tags per photos and the type of tags commonly chosen by users. This study allows us to analyze the behavior of the users related to tagging and draw some important conclusions for our tag recommendation methods.
- We introduce novel tag recommendation methods that take into account also the location of the given photo as well as the location of the existing photos. The key idea of our methods is that not only the similarity in terms of existing tags is important, but also the distance between the existing photos in which the tags appear.
- We implemented a prototype system for location-aware tag recommendations over photos of Flickr and evaluate experimentally our proposed method through examples that demonstrates the effectiveness of location-aware tag recommendation.

The remainder of the chapter is structured as follows. In Section 4.2 we describe our data collections and analyze their properties. Then, Section 4.3 presents an overview of the location-aware tag recommendations system and describes the proposed location-aware tag recommendation methods. In Section 4.4 we evaluate our proposed methods. Finally, in Section 4.5 we discuss related work and in Section 4.6 we provide some concluding remarks.

4.2 Data Collection

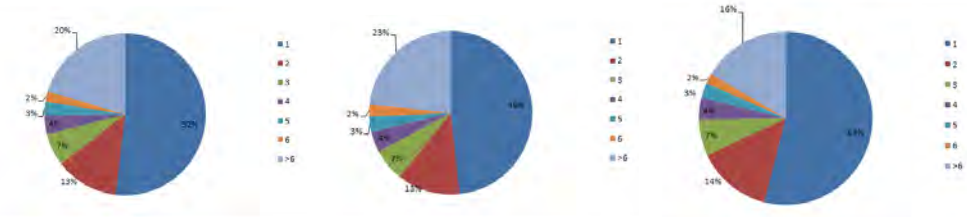
In this section we describe our data collections and provide statistics about the photo tags. In order to design our recommendation strategies it is important to first

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) New York (b) Rome (c) London

Figure 4.2: Tag frequency distribution



(a) New York (b) Rome (c) London

Figure 4.3: Number of tags per photo

study the relevance and quality of the tags. What kind of tags are used for tagging is also important in order to understand which tags are useful for recommendations and how the tags relate to the location of the photo.

We have created three different data collections. Each of them contains 100.000 geotagged photos that are located in New York, Rome and London respectively. Table 4.1 summarizes the number of tags that appear in each collection and the number of unique tags per collection. The collected photos are a random snapshot of the geotagged photos located in the aforementioned cities. For each city the boundary is defined by the bounding box provided at `http://www.flickr.com/places/info/`. The photos were collected between December 2012 and February 2013 and each photo has at least one tag describing it.

Collection	Tags	Unique tags
New York	1.502.454	80.180
Rome	897.185	41.843
London	1.428.047	110.231

Table 4.1: General characteristics per collection.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

4.2.1 Distribution of Tag Frequency

Our data collection of photos collected from Flickr located in New York contains 100.000 photos, with 1.502.454 tags in total, while the unique tags are 80.180. The photo collection of Rome has 897.185 tags in total and the unique tags are 41.843. Finally, the data collection of London has 1.428.047 tags in total and the unique tags are 110.231.

Figure 4.2 shows the distribution of the tag frequency on a log-log scale. The x-axis represent the set of unique tags order based on the frequency in descending order. The y-axis is the tag frequency. We observe that the tag frequency can be modeled by a power law for all data collections.

Tag	Freq.
NYC	47940
New York	45809
NY City	33941
manhattan	27282
NY	26717
USA	15957
City	14637
New	10952
Brooklyn	10741
2012	10126

Table 4.2: New York.

Tag	Freq.
rome	56660
italy	44842
roma	44225
italia	19281
Lazio	8883
2012	8374
Europe	7534
Rom	6917
square	6851
iphoneography	6464

Table 4.3: Rome.

Tag	Freq.
London	68250
UK	30839
England	25760
2012	12459
kenjonbro	11693
trafalgar square	11090
United Kingdom	10023
Westminster	8404
fuji hs10	7981
SW1	7282

Table 4.4: London.

Tables 4.2- 4.4 show the 10 most popular tags for New York, Rome and London respectively. For the New York collection there exist 41.230 tags with tag frequency 1, which are the less popular. To give an example of their relevance we report 10 random of them: walmart, resort, people mover, kristin, bougainvillea, pixie, aviso, World Heritage Site, Beggar, ox. Similarly, for Rome there exists 20.197 tags with frequency 1, while for London there are 59.559 tags with frequency 1.

By observing the distribution of the tags in the each collection, but also by looking at the most popular tags, it is obvious that the most popular tags should be excluded by our recommendation method because these tags are too generic to be helpful for recommendation. Recall that the popular tags include tags such as: NYC, New York, Rome, Italy, London, UK. Similar, the less popular tags with very small frequency (i.e., equal to 1) should be also excluded by our recommendation method, since these tags include words that are misspelled, complex phrases and very specific tags. For example consider the tags: drwho, loo, boring, SF, #noon, dv. Due to their low frequency it is expected that those tags can be

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

useful only in very specific cases and thus are not suitable for recommending to other photos.

4.2.2 Distribution of Number of Tags per Photo

In Figure 4.3 the number of tags per photo are depicted. More precisely, the percentage of photos that have 1, 2, 3, 4, 5, 6, >6 tags for each data collection are depicted. In addition, we consider (Figure 4.4) also the distribution of the number of tags per photo for New York. Figure 4.4 is in log-log scale and the x-axis represents the set of photos ordered based on the number of tags per photo (descending order), while the y-axis refers to the number of tags of each photo. We notice that a high percentage of photos, i.e, approximate 20%, has a high number of tags (more than 6 tags) and there even exist photos with more than 50 tags. Similar results have been also obtained for the other two data collections.

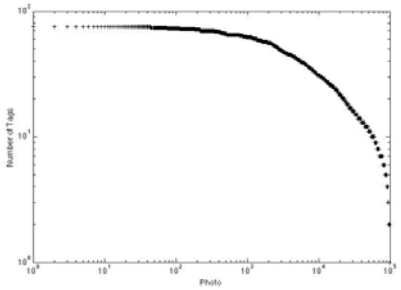


Figure 4.4: Number of tags per photo for New York.

Thus, some photos have a very high number of tags, so that these tags cannot be considered to be representative for the photo. Therefore, our recommendation methods will not use such photos. Moreover, approximately 50% of the photos have only one tag, and again these photos can not be used for tag recommendation that relies on co-occurrence of tags. On the other hand, the fact that a high percentage of photos have only one tag motivates the need for tag recommendation, since all these photos would benefit by a more detailed description.

4.2.3 Analysis based on WordNet

Finally, we analyze what and how users tag by categorizing the tags based on WordNet. We use the broad categories of WordNet and if there exist multiple

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

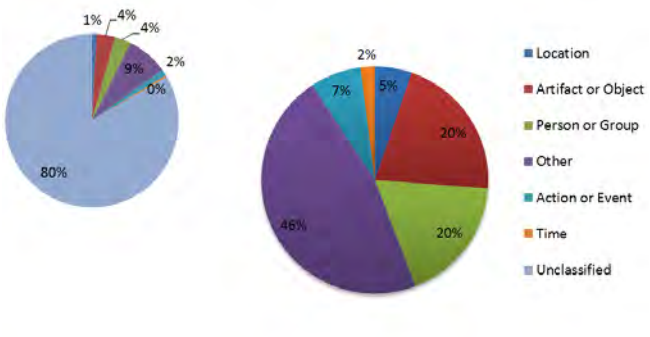


Figure 4.5: Tags per WordNet category for New York.

categories for one tag, this tag is associated with the category of the highest rank. Figure 4.5 presents the distribution of tags for New York based on the most popular categories of WordNet. Following this approach, approximate 20% of the tags can be categorized based on WordNet, leaving 80% of the tags without any category. We depict also in higher details the categorization of the 20% of the tags. By taking into account only the tags that can be categorized, the most frequent categories are ”person or groups” (appr. 20%) and ”artifact or object” (appr. 20%), followed by ”action or event” (appr. 8%), ”location” (appr. 5%), and ”time” (appr. 2%). The category ”Other” (appr. 45%) contains the tags that belong to some category of WordNet, but do not belong to any of the aforementioned categories. We can conclude that the users tag photos not only based on their features, but also based on the information the photo depicts, such as the time taken or the event and the location that is depicted. Similar results hold also for London and Rome data collection.

Since location tags are important, we analyze in more details the location based tags. For the New York data collection it holds that from all unique tags only 777 refer to a location based on WordNet. For the Rome data collection only 411 tags are tags referring to a location based on WordNet, while for London there exist 877 unique location tags. Figure 4.6 depicts the frequency of the location based tags in log-log scale for New York data collection. The x-axis represents the set of unique location tags order based on the frequency in descending order. The y-axis is the tag frequency. We observe that the tag frequency can be modeled by a power law and this holds also for the other data collections.

Tables 4.5-4.7 show the 10 most popular location-based tags for New York, Rome and London respectively. There exist 227 tags with location-based tag fre-

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program ”Education and Lifelong Learning” (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

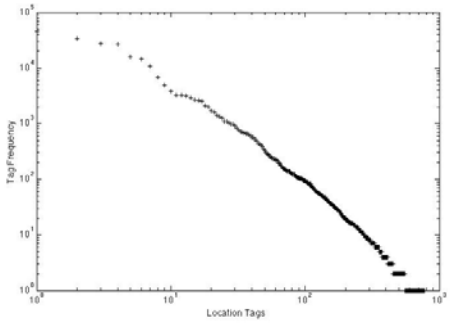


Figure 4.6: Location tag frequency distribution (New York)

Tag	Freq.
New York	45809
New York City	33941
manhattan	27282
NY	26717
USA	15957
City	14637
Brooklyn	10741
United States	6788
america	4853
park	3842

Table 4.5: New York.

Tag	Freq.
rome	56660
italy	44842
italia	19281
Lazio	8883
Vatican City	3067
city	2433
Latium	2002
Piazza	1781
town	604
Umbria	401

Table 4.6: Rome.

Tag	Freq.
London	68250
UK	30839
England	25760
trafalgar square	11090
United Kingdom	10023
Westminster	8404
City	5332
Great Britain	4303
Britain	3870
surrey	2196

Table 4.7: London.

quency 1 for the New York collection. To give an example of their relevance we declare 10 random of them (for the New York collection): vienna, Nepal, Ohio, Bali, Calgary, praia, oslo, Cali, Rio de Janeiro, liverpool, St. Petersburg. Similar for Rome and London there exists 130 and 235 tags with frequency 1. Due to the small number of tags that can be categorized as location tags based on WordNet, but also due their relatively low frequency (i.e., Table 4.6) it is not possible to enhance our recommendation method using the WordNet categories.

4.3 Recommendation Methods

In this section we describe our recommendation methods. The input of our methods is a photo p that is described by a location given by the owner of the photo and a set of tags $\{t_1, t_2, \dots\}$. The goal is to recommend to the use a set of relevant tags $\{t'_1, t'_2, \dots\}$ that could augment the description of p. Our methods rely on *tag co-occurrence*, i.e., the identification of tags frequently used together to annotate

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

a photo. Furthermore, we enhance tag recommendation by taking explicitly into account the location of photos, in order to derive more meaningful co-occurring tags.

4.3.1 System Overview

Figure 4.7 gives a crisp overview of our location-aware tag recommendation system. Our system is built on an existing collection of photos that are geotagged, such as a subset of geotagged photos provided by Flickr. This information is necessary in order to identify frequently occurring tags, as well as to discover keywords that are used together as tags in many photos.

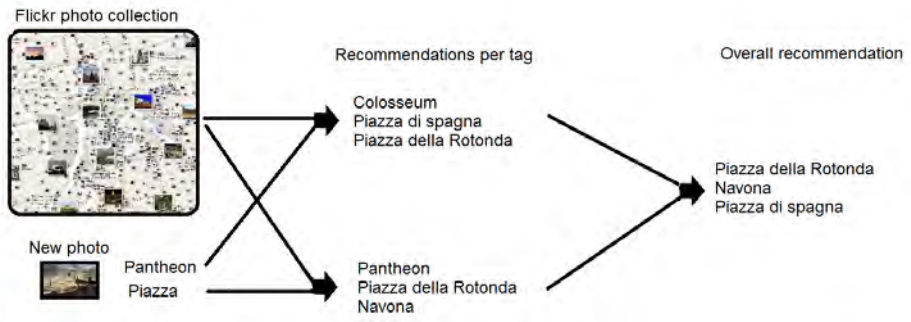


Figure 4.7: System overview.

We adopt a two-phase approach: in the first phase, a set of frequently co-occurring tags is discovered for each input tag $\{t_1, t_2, \dots\}$, while in the second phase, these sets of tags are combined to produce the final tag recommendation. In more details, for each given tag t_i a ranked list of n relevant tags to t_i is retrieved based on the tag co-occurrence and the distance between the given photo and the photos in which the tags co-occur. Each tag is associated with a score that expresses its relevance to given tag t_i. Then, in the second phase, the different lists of relevant tags are combined, by aggregating their partial scores, so that the k most relevant tags are recommended to the user.

Even though different aggregation functions are applicable, we employ a plain strategy of summing the partial scores. Thus, for each tag t'_i, the overall score is defined as the sum of its scores in the ranked lists. Our goal is to produce more qualitative recommendations, by taking into account the location of the photo as well as the location of the existing tags.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union

4.3.2 Tag Recommendation Methods

We employ three different tag recommendation methods: (a) simple tag co-occurrence, (b) range tag co-occurrence, and (c) influence tag co-occurrence. The first method is location-independent and is used as a baseline, while the other two are novel, location-aware methods for tag recommendation.

Simple Tag Co-occurrence Method (Baseline)

The simplest way to measure the relevance of an existing tag to a given tag is tag co-occurrence. Assuming that t_i is the given tag and t_j an existing tag, then we denote $\mathcal{P}_i$ (or $\mathcal{P}_j$) the sets of photos in which tag t_i (or t_j) appear. To compute the co-occurrence of tags t_i and t_j, we need a metric for set similarity. One commonly used metric to express the similarity based on co-occurrence is the Jaccard coefficient, which is defined as the size of the intersection of the two sets divided by the size of their union. Thus, for tags t_i and t_j, the Jaccard similarity is defined as:

$$Jaccard(t_i, t_j) = \frac{|\mathcal{P}_i \cap \mathcal{P}_j|}{|\mathcal{P}_i \cup \mathcal{P}_j|}.$$

Range Tag Co-occurrence Method

One major shortcoming of the simple tag co-occurrence method is that it does not take into account the location of the photo. Intuitively, it is expected that photos that are taken at nearby locations will share common tags, while photos taken far away from each other are less probable to be described by they same tags. This intuition guides the design of both location-aware methods that we propose. Given a radius r and a geo-tagged photo p, we define as $\mathcal{R}(p)$ the set of photos in our data collection that have a distance smaller than r to the location of the given photo p. In other words, photos in the set $\mathcal{R}(p)$ have been geo-tagged with a location that is within distance r from the location of the input photo p. Then, we define a novel measure that combines tag co-occurrence with location information:

$$Range(t_i, t_j) = \frac{|\mathcal{P}_i \cap \mathcal{P}_j \cap \mathcal{R}(p)|}{|\mathcal{P}_i \cup \mathcal{P}_j|}.$$

In this way, for tag co-occurrence, we take into account only the pairs of photos in which both tags appear and are geo-tagged withing a distance r. On the other hand, we divide with the total number of photos in which at least one of the tags

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

appears, thus giving a penalty to tags that appear very often in photos that are distant to each other (i.e., outside the range r).

Influence Tag Co-occurrence Method

One drawback of range tag co-occurrence method is that a radius r needs to be defined as input, and it is not always straightforward how to set an appropriate value, without knowing the distribution of the locations of existing photos. Moreover, the defined range enforces a binary decision to whether a photo will be included or not in the tag co-occurrence computation, based on its distance being above or below the value r. For example, a very small value of radius may result in no photos with the given tag being located into the range, while on the other hand a large radius may result in most (or all) of the photos being located inside the range. Summarizing, the recommended tags are quite sensitive to the value of the radius, which is also hard to define appropriately.

To alleviate this drawback, we propose also a more robust and stable method than the plain range tag co-occurrence method. Given a radius r and a geo-tagged photo p, we define the *influence score* of two tags t_i and t_j as:

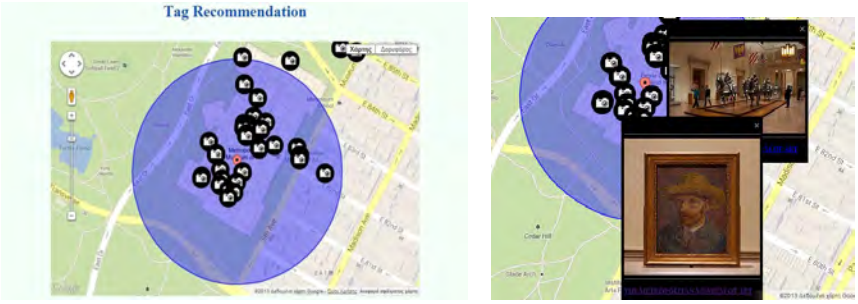
$$infl score(t_i, t_j) = \sum_{p' \in \mathcal{P}_i \cap \mathcal{P}_j} 2^{\frac{-d(p',p)}{r}}$$

, where $d(p', p)$ is the distance between the locations of p and p'. Then the relevance of a given tag t_i and an existing tag t_j is computed as:

$$Influence(t_i, t_j) = \frac{infl score(t_i, t_j)}{|\mathcal{P}_i \cup \mathcal{P}_j|}.$$

The key idea behind the influence score is that tags that co-occur in nearby photos have a higher influence than tags that co-occur in distant photos. This is nicely captured in the above definition by the exponent, which gradually decreases the contribution of any photo p' the further it is located from p. Compared to the range tag co-occurrence method, this method does not enforce a binary decision on whether a photo will contribute or not to the score. Also, even though a radius r still needs to be defined, this practically has a smoothing effect on the influence score (rather than eliminating some photos), thus the score is not very sensitive to the value of r.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.



(a) Recommendation Example

(b) Photos used for Recommendation

Figure 4.8: Example of prototype system.

4.4 Experimental Evaluation

4.4.1 Prototype System

In order to evaluate experimentally our proposed recommendation methods we implemented a prototype system. Our prototype system displays to the user a map by using Google maps and the user can upload a new photo by providing its location (latitude and longitude). Then, in order to use the tag recommendation methods the user is asked to give the radius of interest as well as some initial tags. The recommendation query is posed and the systems displays on the map to the user the location of the new photo, the photos that participate in the recommendation query as well as the recommended tags(Figure 4.8(a)). The user can as depicted in Figure 4.8(b).

In our example, the new photo is uploaded at the location of the Metropolitan Museum of Art in New York (latitude:40.7789 and longitude:-73.9637) and one tag is given by the user namely 'The Metropolitan Museum of Art'. The user decides to use the Influence Recommendation Method and sets the radius to 200 and requests the 3 best matching tags. The recommendation tags are: 'The Met', 'Greek and Roman art' and 'Manhattan'.

4.4.2 Experimental Evaluation

In this section, we provide examples of the proposed recommendation methods of Section 4.3. To this end, we take into account also the conclusions drawn in Section 4.2. Therefore, to avoid tags that are too generic to be helpful for

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union

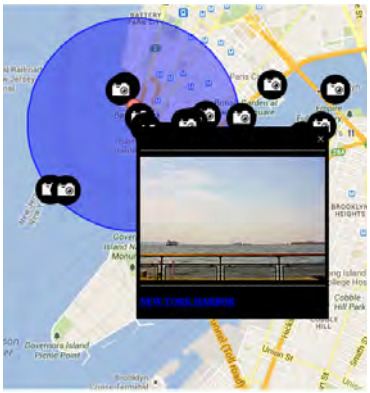


Figure 4.9: Example of recommendation.

Radius	Photos	Range	Influence
500	1098	Frederic Bartholdi, nite, lens adapters	One New York Plaza, Statue of Liberty, Harbor
1000	3828	One New York Plaza, Harbor Statue of Liberty	One New York Plaza, Statue of Liberty, Harbor
1500	6117	One New York Plaza, Harbor, Statue of Liberty	Liberty Island, Statue of Liberty, Harbor
2000	8816	Harbor, One New York Plaza, Statue of Liberty	Liberty Island, Staten Island Ferry, Statue of Liberty

Table 4.8: New York Harbor (Baseline recommends: ”Newtown Creek”, ”Maspeth, New York”, ”DUGABO”).

recommendation, we exclude from the recommendation tags that appear in more than 10% of the photos. Also, we remove from our photo collection photos that have more than 30 tags, as these tags cannot be considered to be representative for the photo. Moreover, photos that have only one tag cannot be used for tag recommendation that rely on co-occurrence of tags, therefore such photos are also removed from the photo collections.

In order to measure the distance between two photos, we convert the longitude and latitude of each photo to the Universal Transverse Mercator (UTM) projected coordinate system. Then, we apply the Euclidean distance in this transformed space.

In our first example we use the New York data collection. Assuming a user

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program ”Education and Lifelong Learning” (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

	Baseline	Range		Influence	
		100	1000	100	1000
1	peeps	Times Square	Times Square	Times Square	Times Square
2	Hood	nikkor 24-70mm f2.8	theatre	lights	theatre
3	Madison Ave	Silver Efex Pro2	Theater District	Theater District	Theater District
4	Lexington Ave	lights	Musical	neon	Musical

Table 4.9: Broadway.

Radius	Photos	Range	Influence
100	219	Musei Vaticani, heritage, DMC-GF1	painting, Musei Vaticani, Vatican Museum
500	11486	Musei Vaticani, Vaticano, Vatican	Musei Vaticani, painting, Vaticano
1000	14450	Musei Vaticani, Vaticano, Vatican	museo, painting, Musei Vaticani, Vaticano
1500	17914	Musei Vaticani, Vaticano, Vatican	museo, Musei Vaticani, Vaticano

Table 4.10: Museum (Baseline recommends: ”museo”, ”Musei Vaticani”, ”sculpture”).

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program ”Education and Lifelong Learning” (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Query	Baseline	Range	Influence
Piazza	Navona, spagna, popolo	pantheon, Rotonda, della	pantheon, Navona, Rotonda
pantheon	colosseum, piazza di spagna, Piazza della Rotonda	Piazza della Rotonda, temple, Dome	Piazza della Rotonda, temple, Dome
Piazza and pantheon	Navona, spagna, popolo	Piazza della Rotonda, temple, Dome	Piazza della Rotonda, temple, Dome

Table 4.11: Rome at Piazza della Rotonda (radius=100).

that uploads to Flickr a photo taken at the Battery Park (40.703294,−74.017411) in the Lower Manhattan of New York. The user gives one tag to the photo namely "New York Harbor". Figure 4.9 shows our prototype system for this query. The recommendation results are shown in Table 4.8. In this example we study how the radius influences our two approaches, while the Baseline fails to recommend relevant tags ("Newtown Creek", "Maspeth, New York" and "DUGABO"). We notice that Range is more sensitive to the radius than Influence. Table 4.8 shows also the number of photos that fall into the region of radius r. This explains the behavior of Range, as for small radius values there exist too few photos to make meaningful recommendations.

Our next example uses again the New York data collection and this time a new photo is located nearby Time Square and the query point location is 40.756116, -73.986409. The given tag by the user is "Broadway". The results are depicted in Table 4.9. In this example, we notice that even for small radius the Influence method manage to retrieve relevant tags, while Range fails for small radius due to the low number of existing photos. On the other hand, both Range and Influence manage to retrieve relevant tags for higher radius values, while Baseline returns more general tags like "Madison Ave".

In the following example we use the Rome data collection. We assume that the given photo is located in Vatican City (query location: 41.903491,12.453214) and it is annotated with the tag "Museum" and the results are shown in Table 4.10. We notice that for small values of radius Range fails to return relevant tags due to the low number of existing photos. On the other hand Influence is influenced by very co-occurred tags like "painting" even for higher radius values, because these

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

two tags appear at many photos together and even if the distance is larger their score is aggregated and alters the final result.

In the next example (Table 4.11) we study the case of a photo that is annotated by 2 tags before the tag recommendation. We use the Rome data collection and we assume that the photo is taken at Piazza della Rotonda in front of Pantheon (41.899134, 12.47681). We set the radius equal to 100 since in the historical center of Rome there are many nearby photos. Location-aware tag recommendation manages to give relevant tags also for generic terms like "Piazza". For "Piazza" and "pantheon" query, the Baseline returns the same results as "Piazza" because there is a higher co-occurrence between this tag and the others, while for the location-aware approaches the results are the same as "pantheon" because there are more photos with this tag nearby the given location.

	Baseline	**Range**	**Influence**
1	hyde	roadrace	the mall
2	Green Park	Piccadilly London	Green Park
3	the mall	Road Race Cycling	st james park'
4	Constitution Hill	the mall	Piccadilly London

Table 4.12: "Buckingham Palace" and "park".

Finally, we examine another example in which 2 tags are given ("Buckingham Palace" and "park"). This time we use the London data collection and the photo is located on the Birdcage Walk in front of the St. James's Park (51.501011, −0.133268). The radius is set to 500 and the results are depicted in Table 4.12. This example tries to illustrate a hard case, as one of the tags (i.e, "Buckingham Palace") is not directly related to the location and the other tag (i.e., "park") is quite generic. We notice that Range fails to return "St. James's Park" as a recommended tag, which is probably the most related term based on the location, but still both Range and Influence manage to recommend more relevant tags than the baseline.

4.5 Related Work

Automatic tag recommendation in social networks has emerged as an interesting research topic recently [43]. Especially in the case of Flickr, tag recommendation

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

has been studied in [42, 21]. In more details, [42] presents different tag recommendation strategies relying on relationships between tags defined by the global co-occurrence metrics. On the other hand, in [21] tag recommendation methods are studied that are personalized and use knowledge about the particular user's tagging behavior in the past. Nevertheless, none of the above methods takes into account the locations of photos. SpiritTagger [33] is a geo-aware tag suggestion tool for photos, but the proposed approach relies on the visual content (such as global color, texture, edge features) of the photo and on the global and local tag distribution. In contrast, our approach takes into account the tag co-occurrence and the distance between the given and the existing photos.

An overview of the field of recommender systems can be found in [1]. A framework that decouples the definition of a recommendation process from its execution and supports flexible recommendations over structured data has been proposed in [27, 28]. Neighborhood-based tag recommendation is studied in [5]. The neighborhood is defined based on a graph and tags are propagated through existing edges.

In [41] the authors also focus on geo-tagged photos and propose methods for placing photos uploaded to Flickr on the World map. These methods rely on the textual annotations provided by the users and predict the single location where the image was taken. This work is motivated by the fact that users spend considerable effort to describe photos [3, 42] with tags and these tags relate to locations where they were taken.

4.6 Conclusions

Tag recommendation is a very important and challenging task, since it helps users to annotate their photos with more meaningful tags, which in turn enables retrieving relevant photos from large photos collections such as Flickr. Nowadays, more and more photos are geotagged, and therefore we investigate how to improve tag recommendation based on the spatial and textual information of the photos. To this end, we analyzed the tags of geotagged photos collected from Flickr and proposed two different location-aware tag recommendation methods. Our experiments show that location-aware tag recommendation is promising and the location of a photo improves the quality of the recommendation. In the future, we aim to investigate in depth how different existing recommendation methods can be improved by combining them with the photo locations.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

4.6.1 Acknowledgments

This research work was developed in cooperation with Ioanna Miliou and the research results where published at:
Ioanna Miliou, Akrivi Vlachou: Location-Aware Tag Recommendations for Flickr, DEXA (1) 2014: 97-104.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

Bibliography

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 17(6):734–749, 2005.

[2] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: recommending advertiser bid phrases for web pages. In *Proc. of WWW*, pages 13–24, 2013.

[3] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, pages 971–980, 2007.

[4] P. Bouros, S. Ge, and N. Mamoulis. Spatio-textual similarity joins. *PVLDB*, 6(1):1–12, 2012.

[5] A. Budura, S. Michel, P. Cudré-Mauroux, and K. Aberer. Neighborhood-based tag prediction. In *Proceedings of Extended Semantic Web Conference (ESWC)*, pages 608–622, 2009.

[6] X. Cao, G. Cong, B. Cui, C. S. Jensen, and Q. Yuan. Approaches to exploring category information for question retrieval in community question-answer archives. *ACM Transactions on Information Systems*, 30(2):7, 2012.

[7] X. Cao, G. Cong, and C. S. Jensen. Retrieving top-k prestige-based relevant spatial web objects. *PVLDB*, 3(1):373–384, 2010.

[8] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *Proc. of SIGMOD*, pages 373–384, 2011.

[9] X. Cao, G. Cong, C. S. Jensen, and M. L. Yiu. Retrieving regions of interest for user exploration. *PVLDB*, 7(9):733–744, 2014.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

[10] Y.-C. Chang, L. D. Bergman, V. Castelli, C.-S. Li, M.-L. Lo, and J. R. Smith. The Onion technique: Indexing for linear optimization queries. In *Proc. of Int. Conference on Management of Data (SIGMOD)*, pages 391–402, 2000.

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

[12] Y. Choi, M. Fontoura, E. Gabrilovich, V. Josifovski, M. R. Mediano, and B. Pang. Using landing pages for sponsored search ad selection. In *Proc. of WWW*, pages 251–260, 2010.

[13] S. Cholette, Ö. Özlük, and M. Parlar. Optimal keyword bids in search-based advertising with stochastic advertisement positions. *J. Optimization Theory and Applications*, 152(1):225–244, 2012.

[14] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[15] Y. Du, D. Zhang, and T. Xia. The optimal location query. In *Proc. of SSTD*, pages 163–180, 2005.

[16] E. Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

[17] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *Proc. of ICDE*, pages 656–665, 2008.

[18] J. Finger and N. Polyzotis. Robust and efficient algorithms for rank join evaluation. In *Proc. of SIGMOD*, pages 415–428, 2009.

[19] A. Fuxman, P. Tsaparas, K. Achan, and R. Agrawal. Using the wisdom of the crowds for keyword generation. In *Proc. of WWW*, pages 61–70, 2008.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Macmillan Higher Education, 1979.

[21] N. Garg and I. Weber. Personalized, interactive tag recommendation for flickr. In *Proceedings of ACM Recommender System Conference (RecSys)*, pages 67–74, 2008.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

[22] V. Hristidis, N. Koudas, and Y. Papakonstantinou. PREFER: a system for the efficient execution of multi-parametric ranked queries. In *Proc. of Int. Conference on Management of Data (SIGMOD)*, pages 259–270, 2001.

[23] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid. Supporting top-k join queries in relational databases. *VLDB J.*, 13(3):207–221, 2004.

[24] I. F. Ilyas, G. Beskales, and M. A. Soliman. A survey of top-*k* query processing techniques in relational database systems. *ACM Comput. Surv.*, 40(4):11:1–11:58, 2008.

[25] I. Kamel and C. Faloutsos. Hilbert R-tree: An improved R-tree using fractals. In *Proc. of VLDB*, pages 500–509, 1994.

[26] F. Korn and S. Muthukrishnan. Influence sets based on reverse nearest neighbor queries. In *Proc. of SIGMOD*, pages 201–212, 2000.

[27] G. Koutrika, B. Bercovitz, and H. Garcia-Molina. Flexrecs: expressing and combining flexible recommendations. In *Proceedings of ACM International Conference on Management of Data (SIGMOD)*, pages 745–758, 2009.

[28] G. Koutrika, B. Bercovitz, R. Ikeda, F. Kaliszan, H. Liou, and H. Garcia-Molina. Flexible recommendations for course planning. In *Proceedings of International Conference on Data Engineering (ICDE)*, pages 1467–1470, 2009.

[29] Z. Li, K. C. K. Lee, B. Zheng, W.-C. Lee, D. L. Lee, and X. Wang. IR-tree: An efficient index for geographic document search. *IEEE TKDE*, 23(4):585–599, 2011.

[30] C.-Y. Lin, J.-L. Koh, and A. L. P. Chen. Determining (k)-most demanding products with maximum expected number of total customers. *IEEE Trans. Knowl. Data Eng.*, 25(8):1732–1747, 2013.

[31] J. Lu, Y. Lu, and G. Cong. Reverse spatial and textual k nearest neighbor search. In *Proc. of SIGMOD*, pages 349–360, 2011.

[32] Y. Lu, J. Lu, G. Cong, W. Wu, and C. Shahabi. Efficient algorithms and cost models for reverse spatial-keyword *k*-nearest neighbor search. *ACM Trans. Database Syst.*, 39(2):13, 2014.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

[33] E. Moxley, J. Kleban, and B. S. Manjunath. Spirittagger: a geo-aware tag suggestion tool mined from flickr. In *Proceedings of Multimedia Information Retrieval*, pages 24–30, 2008.

[34] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley and Sons Ltd., New York, NY, 2000.

[35] P. Papadimitriou, H. Garcia-Molina, A. Dasdan, and S. Kolay. Output URL bidding. *PVLDB*, 4(3):161–172, 2010.

[36] S. Ravi, A. Z. Broder, E. Gabrilovich, V. Josifovski, S. Pandey, and B. Pang. Automatic generation of bid phrases for online advertising. In *Proc. of WSDM*, pages 341–350, 2010.

[37] 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.

[38] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørvåg. Efficient processing of top-k spatial keyword queries. In *Proc. of SSTD*, pages 205–222, 2011.

[39] J. B. Rocha-Junior, A. Vlachou, C. Doulkeridis, and K. Nørvåg. Efficient processing of top-k spatial preference queries. *PVLDB*, 4(2):93–104, 2010.

[40] K. Schnaitter and N. Polyzotis. Optimal algorithms for evaluating rank joins in database systems. *ACM Trans. Database Syst.*, 35(1), 2010.

[41] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *Proceedings of International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 484–491, 2009.

[42] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of International World Wide Web Conference (WWW)*, pages 327–336, 2008.

[43] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1):4, 2011.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

[44] Y. Tao, V. Hristidis, D. Papadias, and Y. Papakonstantinou. Branch-and-bound processing of ranked queries. *Inf. Syst.*, 32(3):424–445, 2007.

[45] G. Valkanas, A. N. Papadopoulos, and D. Gunopulos. SkyDiver: a framework for skyline diversification. In *Proc. of EDBT*, pages 406–417, 2013.

[46] A. Vlachou, C. Doulkeridis, Y. Kotidis, and K. Nørvåg. Reverse top-k queries. In *Proc. of Int. Conf. on Data Engineering (ICDE)*, 2010.

[47] A. Vlachou, C. Doulkeridis, Y. Kotidis, and K. Nørvåg. Monochromatic and bichromatic reverse top-k queries. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1215–1229, 2011.

[48] A. Vlachou, C. Doulkeridis, K. Nørvåg, and Y. Kotidis. Identifying the most influential data objects with reverse top-k queries. *PVLDB*, 3(1-2):364–372, 2010.

[49] A. Vlachou, C. Doulkeridis, K. Nørvåg, and Y. Kotidis. Identifying the most influential data objects with reverse top-k queries. *PVLDB*, 3(1):364–372, 2010.

[50] A. Vlachou, C. Doulkeridis, K. Nørvåg, and Y. Kotidis. Branch-and-bound algorithm for reverse top-k queries. In *Proc. SIGMOD*, pages 481–492, 2013.

[51] D. Wu, M. L. Yiu, G. Cong, and C. S. Jensen. Joint top-k spatial keyword query processing. *IEEE Trans. Knowl. Data Eng.*, 24(10):1889–1903, 2012.

[52] T. Xia, D. Zhang, E. Kanoulas, and Y. Du. On computing top-t most influential spatial sites. In *Proc. of VLDB*, pages 946–957, 2005.

[53] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis. Top-k spatial preference queries. In *Proc. of ICDE*, pages 1076–1085, 2007.

[54] M. L. Yiu, H. Lu, N. Mamoulis, and M. Vaitis. Ranking spatial data by quality preferences. *IEEE TKDE*, 23(3):433–446, 2011.

[55] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. In *Proc. of ICDE*, pages 688–699, 2009.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.

[56] W. Zhang, D. Wang, G.-R. Xue, and H. Zha. Advertising keywords recommendation for short-text web pages using Wikipedia. *ACM TIST*, 3(2):36, 2012.

The research project is implemented within the framework of the Action Supporting Postdoctoral Researchers of the Operational Program "Education and Lifelong Learning" (Actions Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State.