Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-Graduação em Ciência da Computação

EXPLOITING OPEN DATA FOR IMPROVING SPATIAL KEYWORD QUERY APPLICATIONS

João Paulo Dias de Almeida

QUALIFICAÇÃO DE DOUTORADO

Salvador
1 de julho de 2019

JOÃO PAULO DIAS DE ALMEIDA

EXPLOITING OPEN DATA FOR IMPROVING SPATIAL KEYWORD QUERY APPLICATIONS

Esta Qualificação de Doutorado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Frederico Araújo Durão

Salvador
1 de julho de 2019

RESUMO

to do

Palavras-chave: consulta espacial, linked data, LOD, personalização

ABSTRACT

to do

Keywords: spatial query, linked data, LOD, personalization

CONTENTS

Chapter 1—Introduction 1

1.1 Motivation 2
1.1.1 Query results personalization 3
1.2 Objectives of the Proposed Solution 4
1.2.1 Specific Objectives 4
1.2.2 Research Questions 5
1.3 Methodology 5
1.4 Statement of the Contributions 6
1.5 Thesis Structure 6

Chapter 2—Spatial Information Retrieval 9

2.1 Spatial Information Retrieval Systems 9
2.1.1 Spatial Objects 11
2.2 Query 13
2.3 Textual Queries 14
2.3.1 Pre-processing 14
2.3.2 Similarity Measure 15
2.4 Spatial Queries 17
2.4.1 Range Query 17
2.4.2 Spatial Indexes 18
2.5 Preference Queries 20
2.6 Spatial Preference Queries 23
2.7 Top-k Spatial Keyword Query 24
2.7.1 Spatio-textual Indexes 25
2.8 Top-k Spatial Keyword Preference Query 28

Chapter 3—The Semantic Web 31

3.1 Resource Description Framework - RDF 31
3.1.1 Graph Data Model 32
3.2 Ontologies 33
3.3 Linked Open Data - LOD 34
3.4 SPARQL 36
3.5 Summary 36

Chapter 4—Preliminary Proposal to Improve Query Results 37

4.1 Proposed Algorithm 37
4.1.1 Feature Description Enhancement Algorithm 39
4.1.2 Query Result Personalization Algorithm 40
4.2 Related Work 40
4.3 Summary 43

Chapter 5—Preliminary Proposal Evaluation 45

5.1 Datasets 45
5.1.1 Dataset for Experiment 1 46
5.1.2 Dataset for Experiment 2 47
5.2 Methodology 47
5.2.1 User Profiles 48
5.3 Metrics 49
5.4 Experiment 1: Evaluating Query Results 49
5.5 Experiment 2: Evaluating feature selection 52
5.6 Discussion 53
5.6.1 Limitations and Points of Improvements 55

Chapter 6—Final Remarks 57

6.1 Academic Life 57
6.1.1 Completed Classes 57
6.1.2 Publications and Participation in Scientific Conferences 59
6.2 Schedule 60
6.3 Summary 63

Appendix A—User Profile based on Room Aspect Value 73

LIST OF FIGURES

1.1 Points of interest (p) and features (f) associated with their textual descriptions. 3
1.2 Query execution for two different users over points of interest (p) and features (f) associated with their textual descriptions. 4
1.3 Common words in user (U_x) review are used to personalize the query result. 4

2.1 Components of an IR system. Source: Adapted of (BÜTTCHER; CLARKE; CORMACK, 2016). 10
2.2 Basic spatial objects. 12
2.3 A user (blue speech ballon) using speech as input with the Siri interface. Source: (HEARST, 2011). 13
2.4 Textual database *Keeper*. 14
2.5 Inverted File example using existing terms in textual database *keeper*. . . 15
2.6 Textual query execution example using an Inverted File 17
2.7 Spatial selection example: range. 18
2.8 R-tree examples. 19
2.9 AR-tree example. 21
2.10 Spatial Preference queries examples using different ways to define the spatial neighborhood of an interesting object. 23
2.11 Spatial area containing bars and pubs. 25
2.12 Spatial area partitioned into cells using SKIFF 26
2.13 Spatial objects and their respective MBRs. 27
2.14 DIR-tree structure. 27
2.15 *Spatial Inverted Index*. 28
2.16 Interesting objects (p) and features (f) associated with their textual descriptions 29

3.1 An example of an RDF graph describing a person. 32
3.2 An ontology graph representing concepts and relationships between concepts. 33

4.1 Overview of our approach to automatically improve query results. 38

5.1 Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size (k). 50
5.2 Results obtained with RQ and RQ-LD. 51
5.3 Relative NDCG improvements. 51
5.4 SKPQ-LD evaluation using OpinRank. 53

6.1 Ph.D. activities since the course enrollment until thesis defense. 62

LIST OF TABLES

2.1 Dataset example with hotels. 21

5.1 Characteristics of the data obtained at Mapzen. 46

5.2 Example of information available in OpinRank dataset related to the query "great location". 47

5.3 Example of the user profile related to service aspect rating 48

5.4 Score of object p in Traditional SKPQ Compared With the Score Generated by SKPQ-LD, using hotels from Venice. 54

5.5 Score of Object p in Traditional SKPQ Compared With the Score Generated by SKPQ-LD, using hotels from São Paulo. 54

LISTA DE SIGLAS

PGCOMP	Programa de Pós-Graduação em Ciência da Computação	58
PGCA	Programa de Pós-Graduação em Computação Aplicada	58

Chapter 1

INTRODUCTION

The popularization of social networks, mobile applications, and online services contribute to the growth of data available online. Gantz e Reinsel (2012) estimates that by 2020, over 40 trillion gigabytes (more than 5,200 gigabytes per person) of data will have been generated, imitated, and consumed. Within this overwhelming amount of data, there is a type of data called as spatial data. This type of data is information about a physical object that can be represented by numerical values in a geographic coordinate system (also known as spatial object), like latitude and longitude (RIGAUX; SCHOLL; VOISARD, 2001). Spatial data is critical in a large number of application domains like information retrieval, transportation plan, or emergency response. It's been stated that 80% of all business data has some kind of spatial reference (DANGERMOND, 2017).

Location-based services enable their users to describe, rate, and interact with urban spaces. These services manage spatial objects to satisfy the user's information need. In order to access these objects, the Location-based service can employ spatial preference queries. These queries offer a manageable set of "best" answers which satisfy the query best, instead of return a small or possibly huge and unordered result set. Many preference queries specify user preference using query keywords. For instance, a user looking for a Japanese restaurant can specify his preference with the query keywords "japanese restaurant".

This type of preference query defines that the relevant object for the user has a description composed of words that compose the query keywords too (CAO et al., 2012; CONG; JENSEN; WU, 2009). Under those circumstances, the more words in common, the better the object satisfies the user need. However, this evaluation method has limitations, especially to objects with short textual descriptions. For example, suppose a spatial area (e.g., a city) where two spatial objects are located. The query keywords are "japanese restaurant" and each object has one textual description represented by the following strings: "oriental food" and "cinema". This query is not able to return any object because neither the word "japanese" nor "restaurant" is present in any textual description.

A large number of researchers have recently studied how to improve the object's textual description using Linked Open Data (LOD). This improvement is applied in several areas of research, such as Recommender Systems (HEGDE et al., 2011; FERNÁNDEZ-TOBÍAS et al., 2011) and Information Retrieval (KARAM; MELCHIORI, 2013; BECKER; BIZER, 2009). LOD is a set of practices for publishing and connecting structured data (YU, 2011). According to Saquicela et al. (2018), LOD offers high-level information from the data linked in different LOD repositories.

Usually, queries employ a rank to present the best object for the user first. A basic query ranking function is based on the venue popularity, which can be easily estimated by customers review. This approach produces the same rank of objects for two users located in the same place, although they may show distinct preferences. Generic recommendations are useful for taking a glance at the most popular places in a region, but the decision-making process of the users is more complex in general (GASPARETTI, 2017).

Query personalization provides content that is tailored to individuals based on knowledge about their preferences and behavior (HAGEN; MANNING; SOUZA, 1999). Typically, the personalization filters out objects with low value to the user or order them to present data of high value first. In this domain, social network information is valuable. A user profile including the user interests is the main tool exploited to personalize queries. Query personalization systems have been studied in a number of works (MARGARIS; VASSILAKIS; GEORGIADIS, 2018). These works make use of user personal or collaborative preferences that are stored in a preference repository.

1.1 MOTIVATION

Several queries are processed using the Vector Space Model (VSM) to evaluate the textual relevance between query keywords and object's textual description (ALMEIDA; ROCHA-JUNIOR, 2016), (CAO et al., 2012), (CONG; JENSEN; WU, 2009). The VSM indicates that two strings are textual relevant when they share words. The top-k Spatial Keyword Preference Query (SKPQ) is a preference query that uses query keywords to describe the user preference and is processed using VSM. The SKPQ searches for spatial objects of user's interest (points of interest) based on spatio-textual objects[1] of reference (features) in their spatial neighborhood. For example, Figure 2.16 describes a spatial area with spatial objects p (e.g., hotels) and features f (e.g., any establishment). Consider a user interested in book a hotel close to a Japanese restaurant. The user specifies the query keywords "japanese restaurant" and the spatial selection criteria (represented by the circle around the objects p). The SKPQ returns the object p_3 as the best hotel for the user's need, since f_4 has the greatest textual relevance among all features and satisfies the spatial selection criteria. Further details about SKPQ processing are presented in Section 2.2.

Now, suppose a SKPQ with query keywords "oriental food". Considering Figure 1.1, this query does not return any objects. Neither the word "oriental" or "food" are present in any textual description. Note that "oriental food" has semantic relevance to "japanese restaurant", but the evaluation method is not able to identify this relationship. In this example, the query fails to retrieve relevant objects when query keywords are "oriental food". So, we propose a solution using a LOD dataset to enhance the object textual

[1] Spatio-textual object is an object with spatial coordinates (e.g., latitude and longitude) and text.

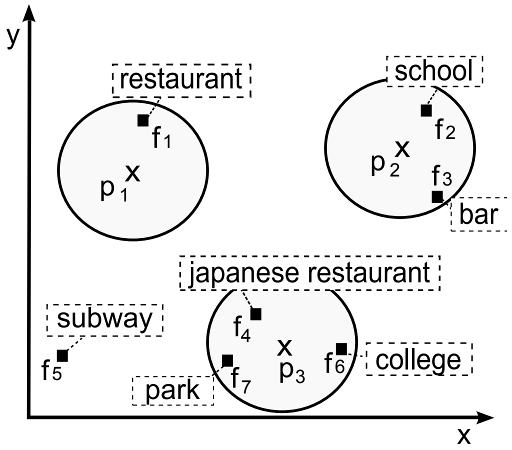


Figure 1.1 Points of interest (p) and features (f) associated with their textual descriptions.

description, in order to achieve better object evaluation. A wider textual description for objects f can improve the object evaluation. If object f_4 had a better textual description, the word "food" or "oriental" might appear in the textual description. In this scenario, the semantic relationship offered by the LOD dataset can be very helpful too.

Motivated by this problem, we use the data available at Linked Open Data (LOD) cloud to enrich the textual description of features. A large number of researches have recently studied how to improve the object's textual description using the LOD cloud. This improvement is applied in several areas of research, such as Recommender (HEGDE et al., 2011; FERNÁNDEZ-TOBÍAS et al., 2011) and Information Retrieval systems (KARAM; MELCHIORI, 2013; BECKER; BIZER, 2009). However, to the best of our knowledge, we are the first to apply a similar improvement in a Spatial Keyword Preference query.

1.1.1 Query results personalization

As described in Figure 1.2, the response provided by the SKPQ is generic and does not consider the user personal preferences. Suppose that two users are looking for a hotel near a japanese restaurant but they have different opinions about what is a good hotel. Unlike user 1, user 2 prefers a comfortable hotel instead of a cheaper one. We can suppose that because looking to user 2 past reviews she gave 5 stars to a hotel describing it as *"The most comfortable hotel I have visited"*. Furthermore, the word "comfort" appears in other high rate reviews from user 2, reinforcing the supposition that she has a personal preference for comfortable hotels.

Aiming further query improvement, we use machine learning to identify which words are common in the user past reviews and relate them to the user interest to personalize the SKPQ. Comparing the user past reviews to a reviews database, it is possible to modify the ranking position of points of interest. Objects whose reviews are similar to the positive reviews made by the user receives a boost in the ranking position as described at Figure 1.3 (or decrease if the reviews are negative). Together with the feature description enhancement, we explore the personalization to improve the accuracy of the SKPQ query

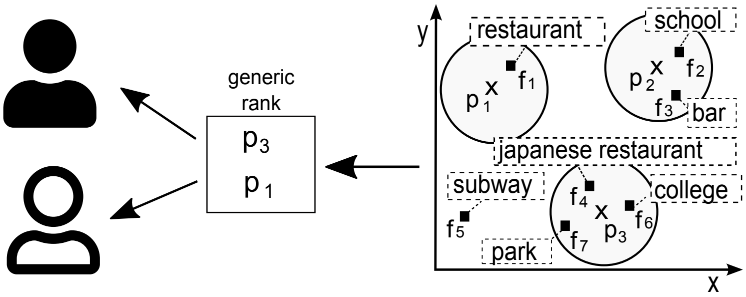


Figure 1.2 Query execution for two different users over points of interest (p) and features (f) associated with their textual descriptions.

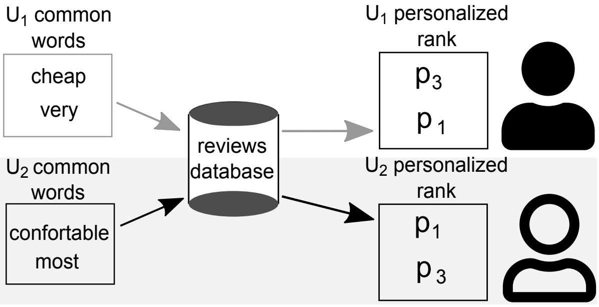


Figure 1.3 Common words in user (U_x) review are used to personalize the query result.

results.

1.2 OBJECTIVES OF THE PROPOSED SOLUTION

This doctoral project aims to improve spatial keyword preference queries towards more accurate query results. Our preliminary solution combines query personalization with the object's description enhancement in order to obtain more accurate queries. This way, the best item for the user is presented first, improving the user experience with the system.

1.2.1 Specific Objectives

- Prepare a literature review to identify the methods applied to improve spatial keyword preference queries.
- Design a query that benefits from Linked Open Data, indicating the application scenarios where this query can be useful.
- Define and implement algorithms to process personalized SKPQ with linked data. We propose a method based on LOD, and point the gaps to further improvements.
- Organize and index data to process spatial queries.
- Conduct experiments to evaluate the query results and compare them with baselines.

1.2.2 Research Questions

The following research questions were proposed conforming to the problems and objectives exposed:

- Q1 - It is possible to improve the SKPQ query results?
- Q2 - It is possible to use Linked Open Data to process SKPQ?
- Q3 - It is possible to combine different techniques to improve SKPQ query results?
- Q4 - How evaluate the proposed approach?

1.3 METHODOLOGY

This work investigates and proposes new methods to improve SKPQ results. The first improvement is a feature description enhancement relying on Linked Open Data. We use SPARQL to access two LOD datasets in order to improve the object's description. The second is a query personalization method using reviews dataset to re-order the query results based on user preference. The results obtained indicate that our approach improves the SKPQ, presenting the best objects at the top positions of the rank. The research protocol is composed of the following steps:

1. **Literature review** - Initially, a literature review was conducted to understand the state-of-the-art for textual improvement using LOD and query personalization. The literature review provides a solid background to the research, presenting varying viewpoint about the research topics.
2. **Mapping research opportunities** - Two techniques to improve the SKPQ were identified using the knowledge obtained from the literature review. Our research suggests that the combination of query personalization with an enhanced object's description would result in a more accurate query result.
3. **Prototype implementation** - A prototype was implemented using the proposed approach. We developed a search engine capable of access LOD datasets and process a spatial keyword preference query.
4. **Experimental evaluation** - Our proposal was evaluated accessing multiple datasets, including a review dataset obtained from TripAdvisor. We employed the metrics described in Section 5.3 to evaluate the query result quality. We did two experiments to evaluate the feature description enhancement, and one to evaluate the query personalization, each with a unique methodology. The query personalization still is a work in progress.
5. **Evaluating the results obtained** - The results were analyzed using NDCG and MAP scores. In addition, we process the SKPQ using real keywords to understand the benefits of the proposed approach in a real scenario.

1.4 STATEMENT OF THE CONTRIBUTIONS

This research aims to contribute to the spatial information retrieval research area in several ways. We describe the contributions in the sequence:

1. **Literature review** - The literature review offers a exploratory study in the spatial information retrieval research area. This study is a critical analysis of the recent researches, coupled with to an investigation of case studies about the discussed problem.

2. **Problem statement** - We investigate the state-of-the-art and identify research topics that researchers could address. Then, we define our research problem and point out solutions based on the literature review.

3. **Algorithms to query processing** - Analyzing the studies conducted in the research area, we propose new algorithms to improve the SKPQ query result.

4. **Proposal evaluation** - In order to evaluate the proposed solution, we conducted an experimental evaluation and compared the results obtained with a baseline. We assess the benefits of exploring Linked Open Data for enriching textual description and evaluate the use of a reviews database to personalize the query.

5. **Problem discussion** - We discuss the key findings of our study, as well as outline directions for future research. We consider the quality of description in the datasets employed and the data integration in the schema-less Semantic Web.

1.5 THESIS STRUCTURE

This chapter introduces the research topic along with the motivation and the possible solutions. Moreover, we expose the objectives and expected contributions of this research. In this section we describe the research design employed in this work. We divide the investigation in three main parts: Theoretical Background, Preliminary Proposal, and Experimental Evaluation.

- **Theoretical Background** - Chapters 2 and 3 presents an overview of the basic concepts that guide this proposal, such as spatial information systems, Resource Description Framework - RDF, and Linked Open Data. In addition, we describe queries similar to the SKPQ and their respective storage indexes. We illustrate each concept with examples to facilitate understanding.

- **Preliminary Proposal** - Chapter 4 define in detail the strategy to improve SKPQ. We present the algorithms to enhance the textual description and to personalize the query. In addition, it encompasses the literature review on textual description enhancement and query personalization.

- **Experimental Evaluation** - We describe the conducted experiment in Chapter 5 together with the methodologies, the datasets, and the results obtained. We explain how the datasets were obtained and its characteristics. Then, we detail the experiment parameters and plot the results in graphs to graphically visualize the results.

Chapter 2

SPATIAL INFORMATION RETRIEVAL

Information retrieval (IR) relates to the representation, search, and manipulation of large collections of unstructured data (BÜTTCHER; CLARKE; CORMACK, 2016; MANNING; RAGHAVAN; SCHÜTZE, 2010). According to Manning et. al. (2010), "unstructured data" refers to data which does not have clear, semantically overt, easy-for-a-computer structure (MANNING; RAGHAVAN; SCHÜTZE, 2010). It is the opposite of structured data, like the data stored in relational databases. Under those circumstances, IR is also used to facilitate "semistructured" search such as finding a document with a specific title and a body containing a specific word.

However, increasing amounts of unstructured data associated with spatial references (e.g., spatial coordinates) are available for indexing and retrieval (PURVES et al., 2018). Spatial Information Retrieval (also known as Geographic Information Retrieval - GIR) (ADAMS, 2018) is the IR field that seeks to develop spatially-aware systems, supporting queries that manipulate spatial coordinates.

Spatial IR systems and services are popular today, helping millions of users worldwide to find information that satisfies their need. Web search engines like Google Maps and Foursquare are examples of spatial IR systems. They help to find people or locations, exhibit user reviews about these locations, and support comparisons such as price comparison between stores products, or distance from the user location to distinct locations of interest.

This chapter introduces the core concepts underlying the topics discussed in this research. We discuss Spatial Information Retrieval Systems followed by spatial objects. Then, we present different types of queries and the indexes used in query processing.

2.1 SPATIAL INFORMATION RETRIEVAL SYSTEMS

A spatial IR system provides access, storagement and management to objects like hotels or restaurants, or text associated with spatial coordinates (e.g., tweets or description of locations). Today, Web Search engines are the most popular IR systems. These systems share a basic architecture and organization that is adapted to the requirements of specific applications. It is important to notice that IR, like any technical field, has words that sometimes differ from their ordinary English meanings (BÜTTCHER; CLARKE; CORMACK, 2016). In this section, we briefly outline the fundamental terminology of the subject in order to provide context for the remainder of the work.

Figure 2.1 illustrates the main components in a spatial IR system. Before conducting a search, a user has a *information need* that drives the search process. (BÜTTCHER; CLARKE; CORMACK, 2016) states that *information need* sometimes refers to a topic, particularly when it is presented in written form. In order to satisfy her information need, the user constructs and issues a query to the IR system. Usually in a Web search[1], the query is composed of two or three terms (FINKELSTEIN et al., 2002; SUGIURA; ETZIONI, 2000). We use "term" instead of "word" because a query term may not be a word. The information need defines if the query term is a date, a number, a phrase, or even a musical note. Wildcard operators may also be included with query terms. For example, "retriev*" might match any wording starting with that prefix (e.g., retrieve, retrieval, retrieves, etc.).

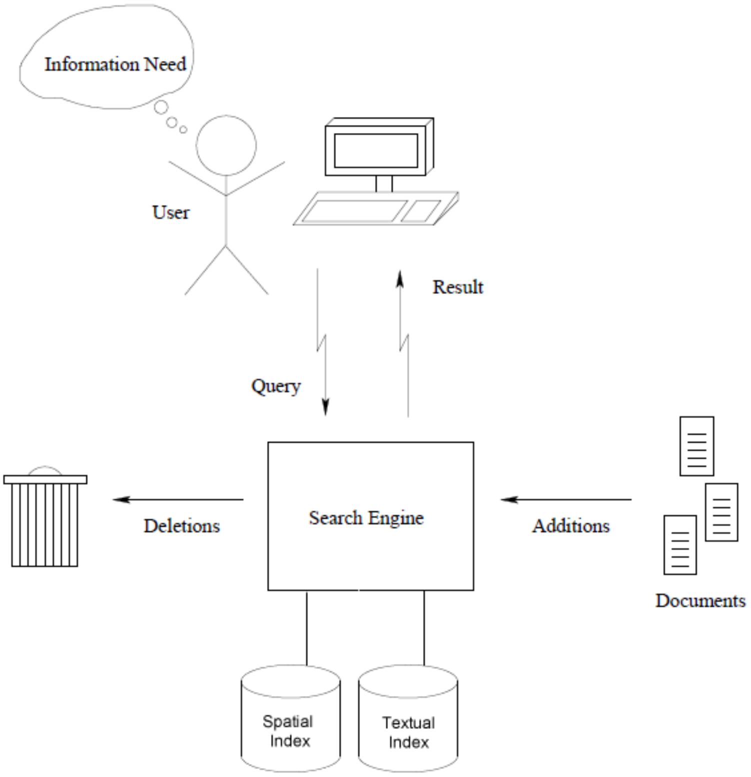


Figure 2.1 Components of an IR system. Source: Adapted of (BÜTTCHER; CLARKE; CORMACK, 2016).

The search engine processes the user's query on the user's local machine or on a cluster of machines in a remote geographic location. For this purpose, it maintains and

[1] The word "search" frequently replaces "(information) retrieval", therefore we use the two synonymously in this work (MANNING; RAGHAVAN; SCHÜTZE, 2010).

manipulates an inverted index for the textual data and a spatial index to spatial data. A Hybrid index can be employed to index the spatial and textual data at the same time. In a nutshell, the inverted index provides a mapping between terms and the locations in the collection in which they occur. The engine uses this index for searching and ranking. The spatial indexes are described at Section 2.4.2 and the hybrid indexes are described at Section 2.7.1. Because of the size of these indexes, efficient algorithms are necessary to access and update them.

The search engine maintains collection statistics associated with the index, such as the number of documents containing each term and the length of each document. These statistics support ranking algorithms. Moreover, the search engine is able to report meaningful results using the original content of the documents (BÜTTCHER; CLARKE; CORMACK, 2016).

In summary, the spatial IR system employs one or many indexes, collection statistics, and other data; to processes the user query and returns a list of results. In order to perform relevance ranking, the search engine computes a score for each document. Then, the system sorts the documents according to their scores and may remove redundant results. According to Büttcher et. al. (2016), a Web search engine might report only one or two results from a single host or domain, benefiting pages from different sources (BÜTTCHER; CLARKE; CORMACK, 2016). The problem of scoring documents with respect to a user's query is one of the most fundamental in the field (LUCCHESE et al., 2016; SKORKOVSKÁ, 2016; YANG; MOFFAT; TURPIN, 2016).

2.1.1 Spatial Objects

A spatial information retrieval system offers support to spatial objects like points, lines, and polygons (GÜTING, 1994; RIGAUX; SCHOLL; VOISARD, 2001). This system provides additional support to spatial data[2] modeling and spatial queries description. In order to manipulate spatial objects efficiently, the spatial database system employs spatial indexes to process the spatial queries (GÜTING, 1994). Spatial indexes are further described in Sections 2.4.2 and 2.7.1.

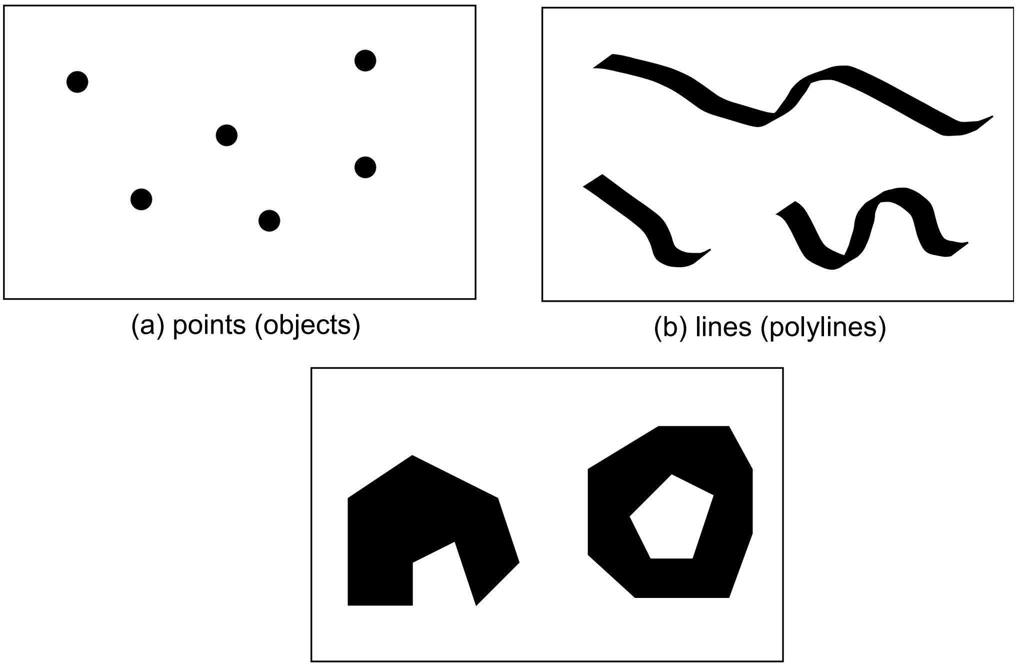
In this work, the space of interest is the Euclidean space R^d, together with the Euclidean distance. Hence, we assume that dimension d is 2. Points are elements of this space. A point has a pair of (Cartesian) coordinates that we denote as x (the abscissa) and y (the ordinate). We focus our attention to the region of R^2 that contains the relevant objects. This region is bounded, and for simplification we assume that it is a sufficiently large rectangle parallel to the axes of the coordinate system. We call it the search space whenever a search operation is to be performed (RIGAUX; SCHOLL; VOISARD, 2001).

A geographic object has two components: (1) a description and (2) a spatial component, also referred to as *spatial object*, which corresponds to the shape and location of the object in the search space. The object is described by a set of *descriptive attributes* (e.g., name and population of a city, or a numeric value indicating the location popularity). Moreover, the spatial object may embody both geometry (location in the

[2] Spatial data is any information associated with geographic coordinates (e.g., latitude and longitude).

underlying geographic space, shape, and others) and topology (spatial relationships existing among objects, such as adjacency, or proximity). The isolated spatial component of a geographic object is the definition of spatial object (LAURINI; THOMPSON, 1992; RIGAUX; SCHOLL; VOISARD, 2001).

The spatial object does not correspond to any standard data type, such as string or integer. The representation of the geometry and topology requires powerful modeling which leads to spatial data models. Usually, the following basic data types are used in spatial data models: point (zero-dimensional object), line (one-dimensional), and region (2D object).



(a) points (objects)

(b) lines (polylines)

(c) regions (polygons)

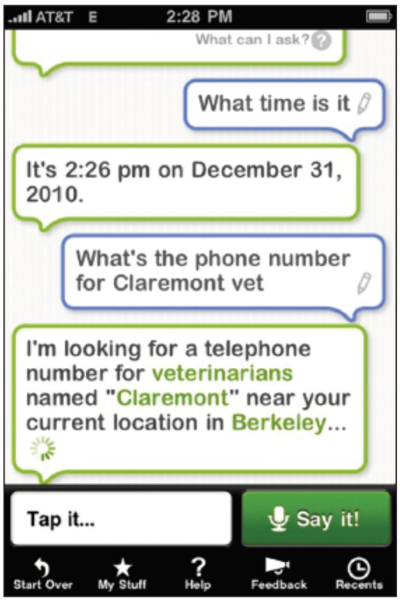
Figure 2.2 Basic spatial objects. Source: Adapted of Rocha-Junior (ROCHA-JUNIOR, 2012).

Figure 2.2 presents some of the basic data types that represents spatial objects: points (objects), lines, and regions (polygons). A point (Figure 2.2(a)) represents an object whose area is not relevant, only its spatial location. For instance, a point can represent a reference object location (i.e., restaurant) or a person location. A line (Figure 2.2(b)) can represent a river, a road, or power lines. Besides, it is important to notice that lines can intersect other lines. At long last, a region (Figure 2.2(c)) usually is modeled like a polygon and can describe spatial objects whose spatial area is relevant like a farm or a forest. Regions are disjoint; however, they can have holes or can be composed of many disjoint pieces (GÜTING, 1994; ROCHA-JUNIOR, 2012).

In this qualification report, the user's point of interest and other points (features) in the search space are "spatial objects". In our scenario, all spatial objects are associated with a textual description. For this reason, we use the term "spatio-textual object" as a synonymous to geographic object as many authors in the literature (BELESIOTIS et al., 2018; CHEN et al., 2017; LIU et al., 2017). Spatial queries are used to efficiently manipulate spatial objects. Nowadays, popular applications such as Google Maps and Booking employs spatial queries to retrieve spatial objects based on the user's query keywords, or object's characteristics (e.g., cheapest hotel in Salvador). We describe spatial queries examples in Section 2.4 to Section 2.8.

2.2 QUERY

In Information Retrieval, a user poses a query to express an information need by converting their desire into language. The language has to fit with the query format employed by the system. Therefore, the system provides to the user an interface where she can use keywords, spatial coordinates, and even speech to define her information need into a query (HEARST, 2011). Figure 2.3 exemplifies a user using speech as an input to a query in Siri[3] application.



What time is it

It's 2:26 pm on December 31, 2010.

What's the phone number for Claremont vet

I'm looking for a telephone number for veterinarians named "Claremont" near your current location in Berkeley...

Figure 2.3 A user (blue speech ballon) using speech as input with the Siri interface. Source: (HEARST, 2011).

Although users typically issue simple queries, IR systems support complex Boolean and pattern matching operators. These facilities may be used to limit a search to a particular web site, to specify constraints on fields such as author and title, or to apply other filters, restricting the search to a subset of the collection. A user interface is the layer between the user and the IR system, simplifying the query-creation process when these richer query facilities are required (BÜTTCHER; CLARKE; CORMACK, 2016).

In addition, users often search for information using an explicit phrase such as "Leonardo da Vinci". In this scenario, the user is interested to find the exact phrase inside a document. Under those circumstances, Boolean operators (AND, OR, and NOT) are used

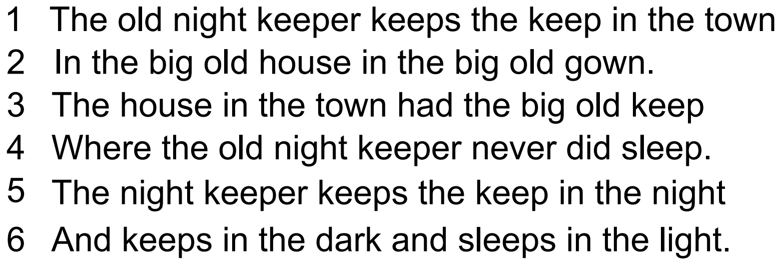
[3] https://www.apple.com/siri/

to combine a set of terms in a query description. For example, the user could define "Leonardo da Vinci" AND sculptures as her query keywords (ZOBEL; MOFFAT, 2006).

2.3 TEXTUAL QUERIES

Textual query is a key technology to search engines (ZOBEL; MOFFAT, 2006). A user can type one or more keywords in a textual query to describe the document she wants to retrieve (MANNING et al., 2008). This query searches and retrieves information from textual collections, returning documents relevant to the user that matches the keyword queries (SALMINEN; TOMPA, 1994). Web search engines (e.g., Google Maps and TripAdvisor) and desktop search systems are examples of daily applications which employ textual queries.

A textual database is a collection of textual data like web pages, encyclopedias, academic publications, or e-mails. Each element from a textual database is called *document*. Accordingly to Manning et al. (2008), *document* is any unit which is chosen to build a Information Retrieval System (MANNING et al., 2008). In a typical Textual Information Retrieval System, a user describes the document she desires using a set of keywords (also known as "bag of words") (ZOBEL; MOFFAT, 2006).



1 The old night keeper keeps the keep in the town
2 In the big old house in the big old gown.
3 The house in the town had the big old keep
4 Where the old night keeper never did sleep.
5 The night keeper keeps the keep in the night
6 And keeps in the dark and sleeps in the light.

Figure 2.4 Textual database *Keeper*. Each text line represents a document. Source: Zobel and Moffat (ZOBEL; MOFFAT, 2006).

For example, based on the textual database described in Figure 2.4, a user can identify a document she is interested in posing a textual query. In this scenario, the system considers each line as a document. Therefore, when a user types a set of keywords in a textual query, this query returns all documents inside the textual database that matches the query keywords. As an example, whether the user types the keyword *big*, the textual query returns the documents 2 and 3, because they contain the keyword.

2.3.1 Pre-processing

Inverted Files (IF) are commonly used to process textual queries efficiently (ZOBEL; MOFFAT, 2006). Create an IF requires to extract the terms from each document in a textual database. For this purpose, a pre-processing stage named *parsing* is applied.

Parsing is realized in two stages: *casefold* and *stop words* removal. The casefold converts every letter in a document to lowercase letters. Applying the casefold in document 1, one obtains "*the old night keeper keeps the keep in the town*" as a result.

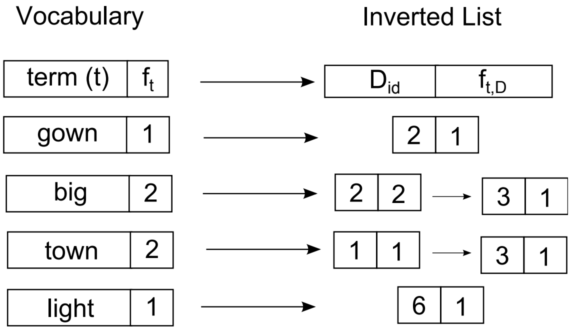


Figure 2.5 Inverted File example using existing terms in textual database *keeper*.

Stop words are those that frequently occur in texts or whose function only is to identify a grammar relationship. For this reason, each language has a specific set of stop words. Removing the stop words and applying the casefold, one obtains the following terms from document 1: "*old night keeper keeps keep town*" (ZOBEL; MOFFAT, 2006). Observe that parsing a document reduces the document size considerably, facilitating the storage and organization process.

The IF is composed of the vocabulary (also known as dictionary of terms) and the set of inverted lists (referred as postings list too). Moreover, each term t in a collection has a corresponding inverted list. This list contains an identifier (D_{id}) for each document (D[4]) that contains the term t in its textual description. D_{id} is followed by a integer value representing the frequency $f_{t,D}$ of a term t occurs in a document's textual description. The vocabulary stores a number f_t of documents which contains the term t and a pointer to the inverted list correspondent to the term t (ZOBEL; MOFFAT, 2006).

For instance, Figure 2.5 illustrates a part of a Inverted File (IF) generated from the textual database *keeper* (Figure 2.4). This IF contains the terms *gown*, *big*. *town*, and *light*. The vocabulary stores terms, the number of documents containing the stored terms, and a pointer to the inverted list related to the term (represented by the unidirectional arrow). The inverted list stores one tuple for each document which contains a term t, this tuple is composed of the document identifier (D_{id}) and the occurrence frequency($f_{t,D}$) of the term t in D.

2.3.2 Similarity Measure

A Textual Information Retrieval System, which employs textual relevance to retrieve documents, uses a *ranking* to order the possible documents to be presented to the user. In order to create a *ranking*, a similarity measure, or heuristic, is applied to indicate the similarity between the document and the query keywords defined by the user (KELES, 2018; MACKENZIE; CHOUDHURY; CULPEPPER, 2015; ZOBEL; MOFFAT, 2006).

The Information Retrieval community widely use the cosine similarity as an effective formulation of the similarity between a document and a set of query keywords (COHEN;

[4] Each line in Figure 2.4 represents a document D while the text inside the line represents the textual description of D.

RAVIKUMAR; FIENBERG, 2003; ZHU et al., 2011; ZOBEL; MOFFAT, 2006). Given a textual query T, composed of a set of terms t ($t \in T$), the cosine similarity $\theta(T, D)$ defines the cosine angle, in a n-dimensional space, between the weight vector[5] and the textual description of document D. As a result, a document D is a possible answer to the user only when exists at least one term $t \in T$ which exists in D too ($\exists t \in T : t \in D$).

Zobel and Mofat (ZOBEL; MOFFAT, 2006) propose the following metrics to calculate the cosine between a document and a query:

- the frequency $f_{t,D}$ of term t in the textual description of D
- the frequency $f_{t,T}$ of term t in the query T
- the number f_t of documents containing the term t
- the total number N of documents in the collection

There are many variations of the cosine similarity formulation (MANNING et al., 2008; ROCHA-JUNIOR, 2012). In this thesis, we use the formulation proposed by Zobel and Mofat (ZOBEL; MOFFAT, 2006), presented in Equation 2.1 which employs the metrics described early.

$$\theta(T, D) = \frac{\sum_{t \in T} w_{t,D} \cdot w_{t,T}}{\sqrt{\sum_{t \in D} (w_{t,D})^2 \cdot \sum_{t \in T} (w_{t,T})^2}} \tag{2.1}$$

The term weight t in document D ($w_{t,D}$) is defined by $w_{t,D} = 1 + ln f_{t,D}$, while the weight $w_{t,T}$ of term t in a query T is $w_{t,T} = ln\Big(1 + \dfrac{N}{f_t}\Big)$. The greater the value of $\theta(T, D)$, the greater is the textual relevance between the document D and the query T. Consequently, $\theta(T, D)$ is also known as the textual score of D related to the query T.

Additionally, $w_{t,T}$ represents a property usually described as *inverse document frequency* (IDF), while $w_{t,D}$ is the *term frequency* (TF). For this reason, the formulation described by the Equation 2.1 is also described in the literature as TFxIDF (ZOBEL; MOFFAT, 2006).

A textual query must generate a *ranking* containing the documents to return to the user. Figure 2.6 exemplifies a textual query processing. Initially, each document has textual score equals zero while a sum array A, of size N, is created to sum the partial textual score of each document. Each array position A_D stores the partial textual score of a document D.

Provided those definitions, each term $t \in T$ contributes $w_{t,D} \cdot w_{t,T}$ (Equation 2.1) to the similarity between a query T and a document D. The A_D position of the sum array (represented by "Adder" in Figure 2.6) stores the summation value obtained from $\sum_{t \in T} w_{t,D} \cdot w_{t,T}$.

The textual score of a document D is the division of its partial score in A_D by the document weight (W_D),

[5] The weight vector is formed by the terms weight t in T.

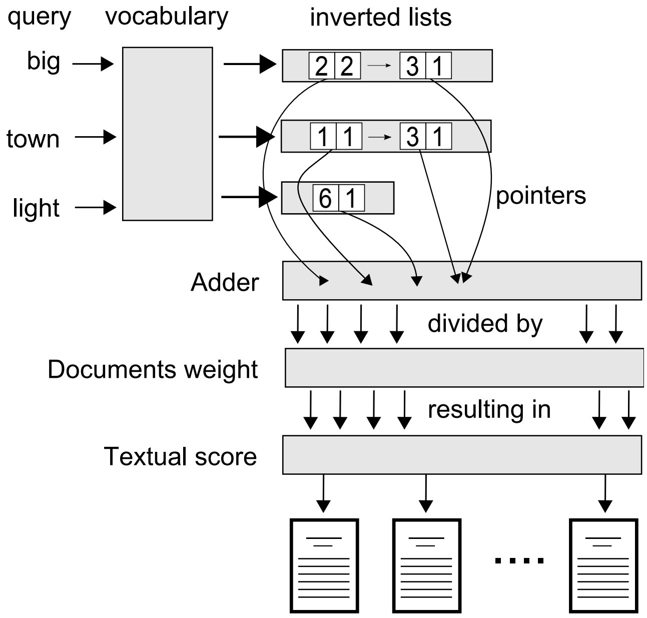


Figure 2.6 Textual query execution example using an Inverted File (IF). Source: Adapted of Zobel and Mofat (ZOBEL; MOFFAT, 2006).

$W_D = \sqrt{\sum_{t\in D}(w_{t,D})^2 \cdot \sum_{t\in T}(w_{t,T})^2}$ (obtained from Equation 2.1).

Last, the documents are ordered by their respective textual scores and then presented to the user.

2.4 SPATIAL QUERIES

Databases adapted themselves to store and organize different types of data efficiently. The spatial data availability associated with technologies advancement made possible a scenario where spatial data is the core of many applications (RIGAUX; SCHOLL; VOISARD, 2001). Today, any individual using a smartphone, is a potential spatial data provider due to the popularization of the Global Positioning System (GPS).

Satellite images, medical equipment, or Geographic Information System (GIS) are other sources of spatial data which provide a large amount of data. Unfortunately, manipulate this volume of data is expensive and impractical to users who don't have proper computational tools. This task becomes even more difficult when is required to analyze the data in details.

2.4.1 Range Query

Because of the large volume of spatial data available for search, the popularity of spatial queries increase. Among the most important types of spatial queries employed in spatial databases are the spatial selections based on predicates (GÜTING, 1994; ROCHA-JUNIOR, 2012). Given a database, a spatial selection returns the set of objects which satisfies the predicate. This predicate can be represented by one or more spatial relationships - the most significant operation provided by the spatial algebra (GÜTING, 1994). These spatial relationships can be topological (i.e. adjacency, disjunction), directional (i.e. above, below, to the left), and metric (i.e. distance), among others. The sentence "*find all restaurant in a 100m radius from my actual location*" is an example of spatial selection.

In this work, we direct our focus to one spatial selection in particular: the *range.*

Range. Given the query location $q.l$ and the distance $dist(p, q.l)$ (euclidian distance between $q.l$ and an object p) , the spatial query *range* retrieve all p objects whose distance values are smaller than the radius r, $dist(p, q.l) \leq r$ (ROCHA-JUNIOR, 2012; YIU et al., 2007). Therefore, r defines the query's spatial neighborhood.

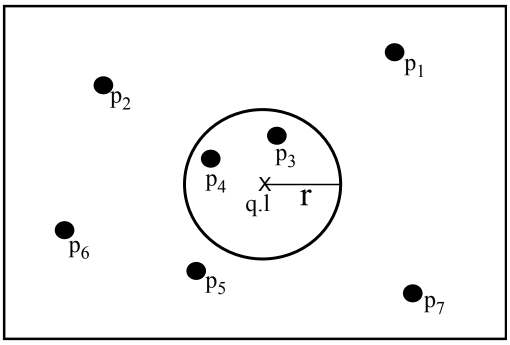


Figure 2.7 Spatial selection example: range. Source: Rocha-Junior (ROCHA-JUNIOR, 2012).

Figure 2.7 illustrates a range query where $q.l$ is the query location and r is the interest radius. Processing this query on the spatial area illustrated in Figure 2.7 returns the points p_3 and p_4 as result.

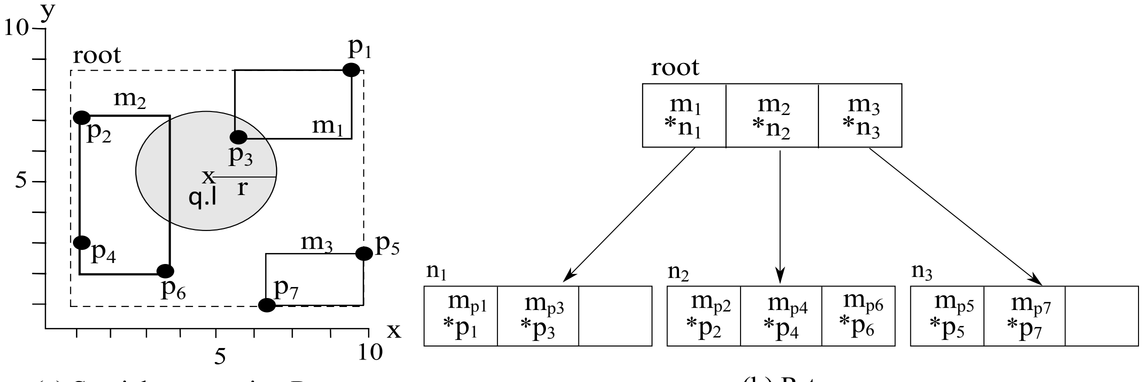
2.4.2 Spatial Indexes

A spatial database system requires a mechanism to improve the spatial objects retrievement, taking in consideration their locations and the user's need (GUTTMAN, 1984). In order to assist in this task, several researchers proposed many spatial indexes (BECKMANN et al., 1990; GUTTMAN, 1984; PAPADIAS et al., 2001; SAMET, 1984). We present some of these spatial indexes in this subsection.

The R-tree is a balanced tree, almost identical to a B-tree (BARUFFOLO, 1999; BAYER; MCCREIGHT, 1970; COMER, 1979) whose leaves have pointers to space-textual objects. R-tree is dynamic; hence, insertion and removal of elements can be performed in conjunction with queries without having to reorganize the tree periodically (GUTTMAN, 1984). In addition, R-tree nodes are generally the size of a disk page, and their structure is designed to search only a small number of nodes. Thus, each node of the R-tree has a minimum and a maximum number of entries (GUTTMAN, 1984; ROCHA-JUNIOR, 2012).

There are two types of nodes in an R-tree: intermediate nodes and leaf nodes. The intermediate node contains pointers to the descendant nodes, while the leaf nodes have pointers to the indexed objects. The entries of an R-tree are formed by (MBR, *id*). Minimum Bounding Rectangle (MBR) is an n-dimensional rectangle surrounding the indexed object, and *id* is a number that identifies the input. The *id* of an intermediate node is a pointer (address) to another node in the tree (descendant node), while the MBR of an intermediate entry involves the MBRs of all entries in the child node. In the input of a leaf node, *id* is the identification of the object in the database and the MBR is the smallest possible n-dimensional rectangle that can wrap the indexed spatial object (ROCHA-JUNIOR, 2012).

Figure 2.8(a) is the representation of a spatial area where objects (p) are indexed in a R-tree. Under those circumstances, $q.l$ is the query location, r defines the spatial neighborhood of $q.l$, and m_1, m_2, m_3, and *root* are the MBRs. On the side, in Figure 2.8(b), the root is a intermediate node that has three intermediate entries m_1, m_2, m_3 which point to the leaf nodes n_1, n_2, n_3, respectively. The intermediate entry $(m_1, *n_1)$ contains the MBR m_1 that involves all stored objects in node n_1, and a pointer $*n_1$ pointing to the node n_1. The leaf node n_1 contains two leaf entries: $(m_{p1}, *p_1)$ and $(m_{p3}, *p_3)$, where m_{p1} is the MBR involving the spatial object p_1 and $*p_1$ is the pointer (identifier) to object p_1 in the database.



(a) Spatial query using R-tree

(b) R-tree

Figure 2.8 R-tree examples. Adapted of Rocha-Junior (ROCHA-JUNIOR, 2012).

For example, Figure 2.8(a) presents a range query processed with a R-tree. The query searches for spatial objects inside the spatial neighborhood defined by r. In other words, the query searches for objects inside the circumference which has 'x' as the center and r as the radius. The range query starts the search in the root and then searches the entries, verifying which entry has a MBR distance to $q.l$ smaller than the size of r. Knowing that p is the nearest point in a MBR to $q.l$, $dist(p, q.l)$ defines the shortest distance of a MBR to $q.l$, this way $dist(p, q.l)$ have to be lower than r to the entry be visited. In Figure 2.8(a), two entries satisfy this condition: $(m_1, *n_1)$ e $(m_2, *n_2)$. As a result, the leaf nodes n_1 and n_2 are accessed to search for the leaf entries whose MBR[6] is inside the spatial neighborhood defined by r, returning object p_3 as consequence.

[6] In this case, the leaf entries are bi-dimensional points. Thereby, the upper right vertex is identical to the lower left vertex.

The R-tree is based on a heuristic optimization, consisting in minimize the MBR area of each intermediate node. However, this criterion proved not to be the best (BECKMANN et al., 1990). One of the most well-known variations of the R-tree is the R*-tree (CHEN et al., 2013; HARIHARAN et al., 2007; WU et al., 2012; ZHOU et al., 2005). The R*-tree is superior to the R-tree in query processing and in the algorithm that defines the MBR of the nodes (BECKMANN et al., 1990).

The R*-tree reduces the coverage area of the MBRs involving intermediate nodes. Thus, fewer tree branches are used during query processing, resulting in less access to disk pages. In addition, R*-tree reduces the overlap between MBRs, reducing the probability of having more than one MBR covering the same area and increasing the efficiency of the query (ROCHA-JUNIOR, 2012).

Another widely used variation of R-tree is the aggregate R-tree (aR-tree), proposed by (PAPADIAS et al., 2001). The main feature of aR-tree is to use pre-aggregated non-spatial data to optimize query processing. In other words, each node of an aR-tree has a non-spatial data (eg, a numeric value) added.

For instance, assume that each object p in Figure 2.8 has a non-spatial score (numeric value). In this context, a query can be made to search for objects in the spatial neighborhood defined by r and that have a score greater than 0.7. In a traditional R-tree, this query needs to be performed in two steps. Initially, all objects that are in the spatial neighborhood of $q.l$ are selected. Then the score of each selected object is checked, and only those that have a score greater than 0.7 are returned (ROCHA-JUNIOR, 2012; PAPADIAS et al., 2001).

In order to optimize this process, each intermediate node of aR-tree stores a value that is obtained by an aggregated function applied to the child node inputs. Under those circumstances, the Max aR-tree is used because it employs the aggregated function $max()$. Thus, the maximum value of the score on the child nodes is added to their respective intermediate node (ROCHA-JUNIOR, 2012; PAPADIAS et al., 2001).

Figure 2.9 represents a Max aR-tree where the aggregated function $max()$ was applied. Therefore, it is observed that the score stored at the intermediate input $(m_1, 0, 9, *n_1)$ is 0.9 because this is the highest score value between the entries of the node n_1: $(m_{p1}, 0, 5, *p_1)$ e $(m_{p3}, 0, 9, *p_3)$. The structure and the way the aR-tree query is executed are similar to that of the R-tree. However, only entries that satisfy the spatial and non-spatial conditions are visited. For example, to find objects that are in the spatial neighborhood of $q.l$ and have a score greater than 0.7, the root is accessed for the input that satisfies these two conditions (neighborhood criterion and score). Thus, only the input $(m_1, 0, 9, *n_1)$ is visited, and the object p_3 is returned because it is the only one that has a score greater than 0.7 and has a distance to $q.l$ lower than the size of r.

2.5 PREFERENCE QUERIES

Databases provide a rigid way to define the characteristics of the retrieved data while using traditional queries (LACROIX; LAVENCY, 1987). The lack of flexibility culminate in a very large, or very small, set of retrieved data. Therefore, current Information Systems employ techniques to describe and process user preferences (CHOMICKI, 2003).

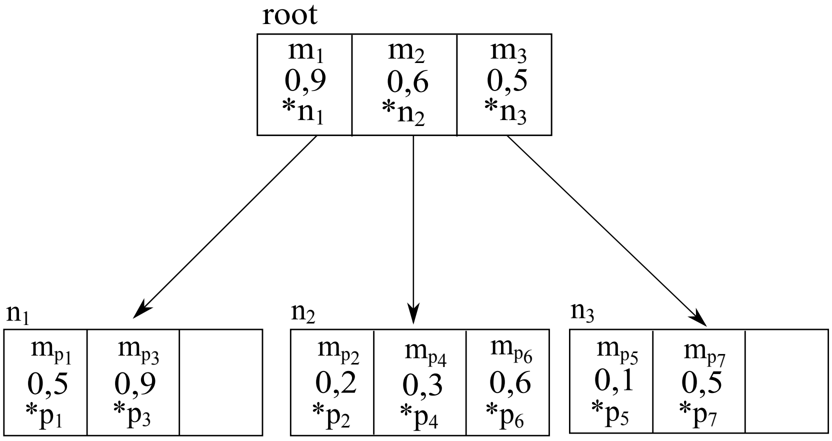


Figure 2.9 AR-tree example.

These preferences are a important tool to filter the information, reducing the data volume presented to the user (CHOMICKI, 2003).

In detail, Table 2.1 illustrates a dataset H which contains information about hotels, such as their respective daily price; and the distance from these hotels to the beach. Assume a user who does not know the dataset and wants to find a cheap hotel. Using traditional queries, the user can request to list the hotels with daily price below 60. In this case, no hotel will be returned. In contrast, if the user requests the list of hotels with rate higher than 60, the complete dataset will be returned. In this way, the user will have to go through the database until she finds the hotel she wants, making it difficult to find the cheapest hotel.

Table 2.1 Dataset example with hotels. Each object (hotel) contains two attributes: *price* (daily price in US dollars) and `distance` (distance to the beach in meters). Source: Adapted of Rocha-Junior (ROCHA-JUNIOR, 2012).

Hotel	**Price** ($)	**Distance** (m)
h_1	300	50
h_2	100	100
h_3	500	100
h_4	90	300
h_5	250	500

Preference Queries (LACROIX; LAVENCY, 1987) allow the user to express their preferences more clearly and accurately[7]. One can solve the problem described above by setting the query as follows: "select hotels with the lowest price values, stop after k"

[7] Borzsony, Kossmann and Stocker (BORZSONY; KOSSMANN; STOCKER, 2001) demonstrate how to implement a preference query using SQL (without making modifications in the database system). They discuss the reasons to such an implementation presents poor performance when compared to an implementation of the preference query using an extension of a database system with a new logical operator representing the preference query.

(ROCHA-JUNIOR, 2012). Thus, by considering $k = 3$ and using the data from Table 2.1, this preference query returns the hotels h_2, h_4, and h_5.

Preference queries are classified by the methods employed to express the users' information need. The *qualitative* preference query specifies the user preference directly between pairs of objects (tuples) in the database, using a preference formula $f(a, b)$. Given two objects h_1 and h_2 in dataset H, the preference formula $f(h_1, h_2)$ determines whether an object satisfies the user's needs. The preference formula $f(h_1, h_2)$ is a binary operation between objects h_1 and h_2. Thus, when the result of this formula is *true*, the query considers object h_1 satisfy the user's needs better than the object h_2. The preference formula is defined using logical operators (CHOMICKI, 2003; KIESSLING, 2002).

For example, consider the database described by Table 2.1 and a user interested in the cheapest and closest to the beach hotel. This user interest can be described by the preference formula $f_1(a, b) = [(a[\text{rate}] \leq b[\text{rate}]) \wedge (a[\text{distance}] \leq b[\text{distance}]$. In Table 2.1, the object h_2 satisfies the user's better than the object h_3. Both hotels have the same distance to the beach, however the hotel h_2 is cheaper. In this scenario, we say that h_3 is dominated by h_2 since $f_1(h_2, h_3) = \mathit{true}$. All not dominated objects are valid responses to the query described by the formula $f_1(a, b)$.

On the other hand, the *quantitative* preference query specifies the preference indirectly for each object in the data set. A score function evaluates the attributes of an object, producing a numeric value (score) that represents the importance of this object to the user's needs. Quantitative queries are often referred to as *top-k queries.* This type of query requires a function to calculate the objects' score and the number of objects (k) to return from the database (ROCHA-JUNIOR, 2012).

For instance, in the dataset H presented in Table 2.1,the hotel h_1 can be represented by $h_1 = \{300, 50\}$, where the value 300 is positioned in column (dimension) 1 of the table and the value 50 in column 2. One can use the quantitative preference query to find the three cheapest and closest hotels to the beach. Assuming a score function $f(h) = 0.5 * h[\text{rate}] + 0,5 * h[\text{distance}]$, objects with lower scores are those closer to the user's need. Thus, the scoring function returns the score values $f(h_2)$ = *100*, $f(h_1)$=*175*, and $f(h_3)$=*195* for the objects h_2, h_1, and h_3, respectively. Therefore, the quantitative preference query returns the objects h_2, h_1 e h_3 as response.

Most top-k query processing techniques use scoring functions called monotonic functions, since these functions have special properties that allow efficient processing of top-k query (ILYAS; BESKALES; SOLIMAN, 2008). Consider an object $h \in H$ represented by $h = h[1], ..., h[n]$, where $h[i]$ is a numerical value in the i dimension. A function f_h defined on the attributes (dimensions) of an object h is monotonic, if for all objects $h, q \in H$, $f_h \leq f_q$ when $h[i] \leq q[i]$ for all i (ROCHA-JUNIOR, 2012). To demonstrate, the function $f(h) = 0,5 * h[\text{price}] + 0,5 * h[\text{distance}]$ would be considered monotonous if for every object $h_x, h_y \in H$, $f(h_x) \leq f(h_y)$ when $h_x[i] \leq h_y[i]$. Since $f(h_2) \leq f(h_1)$ ($f(h_2) = 100$, and $f(h_1) = 175$) but $h_2[2] > h_1[2]$ ($h_2[\mathit{distance}] = 100$, and $h_1[\mathit{distance}] = 50$), this function is not monotonic.

2.6 SPATIAL PREFERENCE QUERIES

Spatial databases manage large collections of geographic entities. Each entity has geographic coordinates which indicate the position of the object in space. Moreover, it is common to associate non-spatial information to the geographic coordinates such as textual description, object name, size, or price of the object (YIU et al., 2007).

Top-k spatial queries return a set of spatial objects (geographic entities) that can serve the user's need. However, each query defines its own set of parameters to represent the user preference. Yiu et al. (YIU et al., 2007) present a new type of query top-k, the Spatial Preference Query. In this query, the k best user's interesting objects are defined through the quality of features[8] in the spatial neighborhood of each interesting object.

Thereby, given a set of interesting objects P (e.g., candidate locations), the Spatial Preference Query returns the k objects in P with the highest scores. The score of an interesting object is defined by the quality of the feature (e.g., cafes, restaurants, hospitals) in its neighborhood. In this fashion, the feature quality can be obtained through an online ranking system, such as Booking[9] or Foursquare[10] where users evaluate various types of features (YIU et al., 2007).

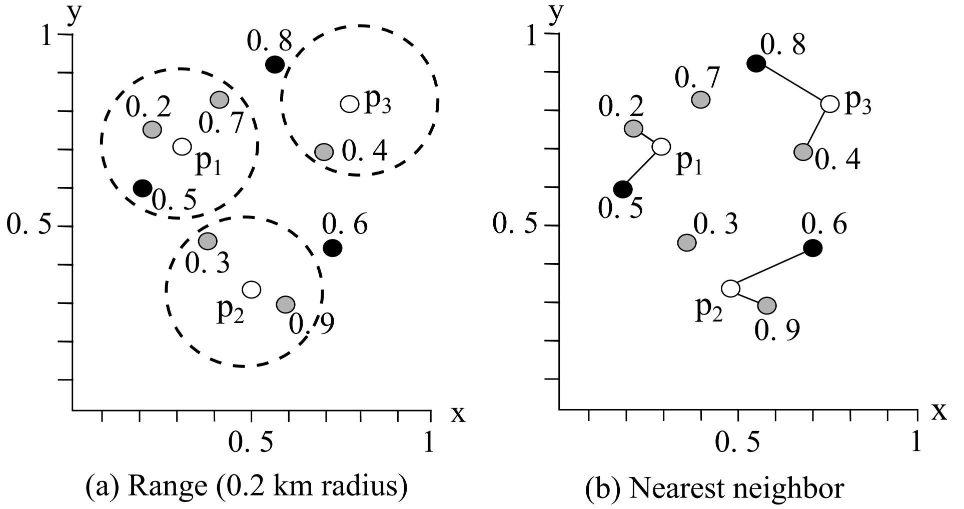


Figure 2.10 Spatial Preference queries examples using different ways to define the spatial neighborhood of a point of interest. Source: Yiu et al. (YIU et al., 2007).

For example, the white points p in Figure 2.10 represent interesting objects. In addition, the gray dots represent restaurants while the black dots represent cafeterias. Each restaurant and cafeteria (black and gray dots) has a predefined score value, represented by the real number positioned around each of these points. The bigger the feature score, the higher the feature quality.

[8] Consider "feature" as a class of objects in a spatial map, such as a specific installation or a service. Each feature is associated with a score that is predefined by a classification system.

[9] www.booking.com

[10] www.foursquare.com

Assuming a tourist wants to get the best hotels in terms of cafeterias and restaurants, the Spatial Preference Query returns the interesting objects (hotels) with the highest scores. In other words, the tourist is interested in the hotel p that maximizes the score $\tau(p)$, defined as the sum of maximum restaurant quality and maximum cafeteria quality in the neighborhood of p (i.e., the dotted circle at p with a 0.2 km radius). Thus, the interesting objects score values for this range query are $\tau(p_1) = 0.7 + 0.5 = 1.2$, $\tau(p_2) = 0.9 + 0 = 0.9$, and $\tau(p_3) = 0.4 + 0 = 0.4$. Interesting objects that do not have cafeterias or restaurants in their neighborhood receive the value zero as the score, situation represented by objects p_2 and p_3. As can be seen, we obtain the object p_1 as the top-1 result of the range Spatial Preference query (YIU et al., 2007).

Likewise, Figure 2.10 (b) illustrates the scenario where the score $\tau(p)$ of a hotel is taken as the sum scores of its nearest restaurant and cafeteria (indicated by connecting line segments). Therefore, we have $\tau(p_1) = 0.2 + 0.5 = 0.7$, $\tau(p_2) = 0.9 + 0.6 = 1.5$, $\tau(p_3) = 0.4 + 0.8 = 1.2$, resulting in p_2 as the best hotel (YIU et al., 2007).

Generally speaking, the Spatial Preference Query uses two steps to select interesting objects. First, it calculates the distance of the interesting object for a given feature. Then, it orders the objects by an aggregation function on their scores (YIU et al., 2007).

Besides this Top-k query, a lot of work is being developed in this research area (LI et al., 2018; SHANBHAG; PIRK; MADDEN, 2018; CARMEL; GUETA; BORTNIKOV, 2018; MENG; ZHANG; ZHAO, 2018), demonstrating the popularity of Top-k queries.

2.7 TOP-K SPATIAL KEYWORD QUERY

Among spatial queries, there those that use keywords to express the user's information need. In this section, we will describe some queries that use this model to retrieve the desired information. Then, we discuss hybrid indexes capable of simultaneously indexing spatial and textual data. These spatio-textual indexes aim to support efficient processing of queries which access data with spatial and textual properties.

Given a spatial location and a set of keywords, a *top-k Spatial Keyword* query (SK) (CAO et al., 2012; CHEN et al., 2013) returns objects that are spatially close to the user's location and textually relevant to the keywords. All returned objects have these two characteristics: user proximity and textual relevance. A score function evaluates the spatial proximity between an object and the user, as well as the textual relevance of the object description considering the set of keywords. The response of this query is ordered from the score values generated for each object by the scoring function.

Suppose a user wants to find a bar where they have samba presentation, in the spatial area described by Figure 2.11. This user poses a top-3 spatial keyword query q with the following keywords: "samba" and "bar". The user location is the same query location q in Figure 2.11.

In this example, the top-k Spatial Keyword query returns a ordered set containing the objects p_4, p_6, p_7. The object p_4 is the top-1 result because its textual description is similar to the keywords provided by the user, and it is the object closest to the query location. Following, object p_6 is the top-2 result, since the textual description of p_6 is more relevant than that of p_7, and p_6 also gets closer to q than p_7.

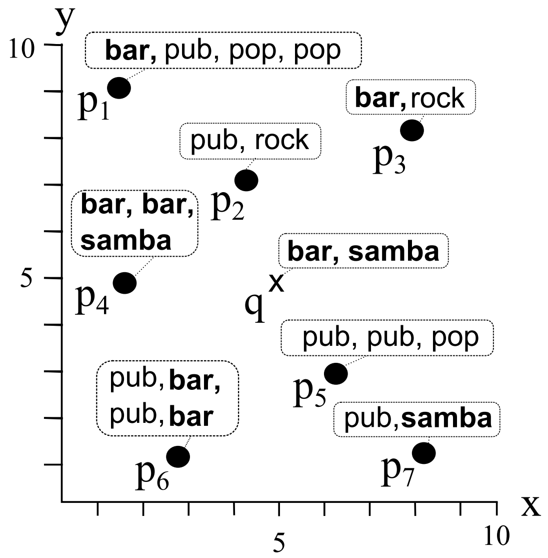


Figure 2.11 Spatial area containing bars and pubs. Source: Rocha-Junior et al. (ROCHA-JUNIOR et al., 2011).

2.7.1 Spatio-textual Indexes

Many applications now use a large amount of spatial data, such as Twitter[11] and Flickr[12]. These applications can benefit from Spatial Keyword Query (SK) and other spatio-textual queries, but the cost of processing these queries is prohibitive (ROCHA-JUNIOR et al., 2011). For this reason, spatio-textual indexes play an important role in the processing of these queries. These indexes store data that contains textual and geographic information, enabling efficient processing of spatio-textual queries (CHEN et al., 2013).

Spatial-Keyword Inverted File (SKIF) proposed by Khodaei, Shahabi and Li (KHODAEI; SHAHABI; LI, 2010) is an Inverted File (IF) capable of indexing and searching for spatial and textual data in an integrated way, using only one structure to manage the two parts of data simultaneously. Figure 2.12 illustrates a SKIF where the space is partitioned into cells. SKIF represents the keywords as these cells.

Similar to IF, SKIF is composed of a vocabulary and a set of inverted lists. The vocabulary contains all terms in objects' textual descriptions and identifiers for the cells that constitute the grid over the spatial area of interest. For each distinct term, the following values are stored in the vocabulary: the number of objects p_t containing the term t, an inverted list pointer, and the indexed term type. Each term t has a corresponding inverted list, where is stored the objects identifiers that have the term t and the normalized frequency with which term t appears in each object description (KHODAEI; SHAHABI; LI, 2010).

SKIF is designed to process a query capable of returning the k objects that have the highest textual and spatial scores concomitantly. However, SKIF process the location

[11] www.twitter.com

[12] www.flickr.com

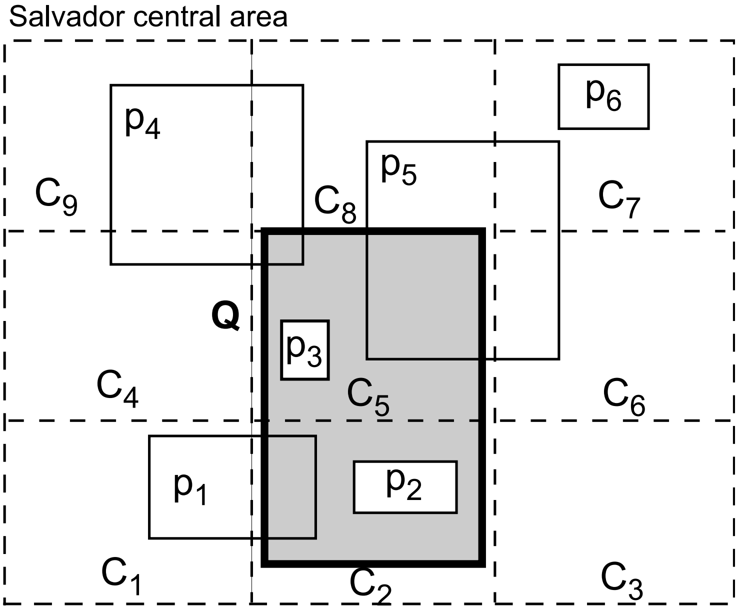


Figure 2.12 The search area is defined by the user (e.g., "central area of Salvador"), then the system divides it into a grid (C_i) to apply the SKIF index. The dark rectangle represents the query location, while light rectangles (p_i) represents objects near the user's search area. Source: adapted of (KHODAEI; SHAHABI; LI, 2010).

of each object as a region rather than a point. Spatial relevance is expressed by the overlap between the query region (dark region) and the region of an object (p_i) (CHEN et al., 2013). Therefore, despite being a hybrid index, SKIF is not able to process a top-k Spatial Keyword query because it does not consider the query location as a point. Even though Chen et al. (CHEN et al., 2013) modified SKIF to process Boolean Range queries (BRQ), they report not be able to identify a way to process top-k queries using SKIF.

Together with SKIF, many other cell-based structures have already been proposed to process spatio-textual objects (BENTLEY; FRIEDMAN, 1979; GUTTMAN; STONEBRAKER, 1982). We discuss below other hybrid indexes capable of indexing spatio-textual objects. These indexes employ trees such as the R-tree proposed by Guttman (GUTTMAN, 1984), and the R*-tree proposed by Beckmann et al. (BECKMANN et al., 1990).

Cong, Jensen e Wu (CONG; JENSEN; WU, 2009) incorporate document similarity to propose a new spatio-textual index called DIR-tree (Document-similarity enhanced Inverted file R-tree). DIR-tree combines spatial and text information during the index building process, keeping at the same level of the tree objects that are spatially close to each other. In addition, it maximizes the textual similarity between objects that are part of the same MBR (CONG; JENSEN; WU, 2009; WU; CONG; JENSEN, 2012). A β parameter is introduced to determine the weight between the spatial and textual part of the data. For example, depending on the β value, the objects of the same MBR can be close but with little textual relevance to each other.

Each DIR-tree node is associated with an Inverted File (IF), and each leaf node contains an object summary that provides textual information about the objects (LI et al., 2011). Figure 2.13 presents spatial objects P_i and their respective MBRs R_i. Coupled with Figure 2.14 that exemplifies the organization of a DIR-tree to index the objects in Figure 2.13

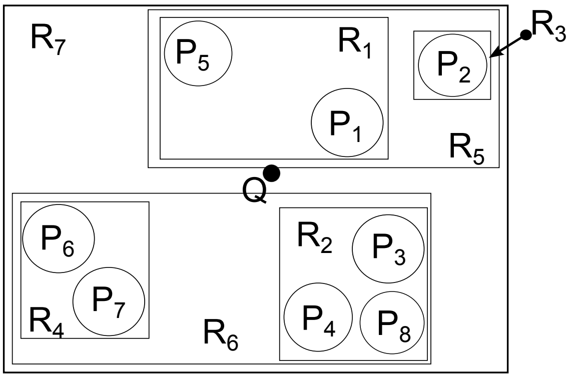


Figure 2.13 Spatial objects P_i and their respective MBRs R_i. Source: (CONG; JENSEN; WU, 2009).

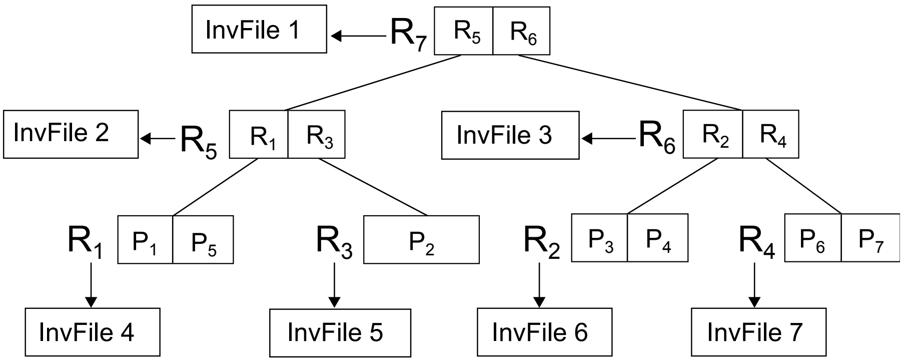


Figure 2.14 DIR-tree structure. Source: (CONG; JENSEN; WU, 2009).

To demonstrate, consider a spatio-textual query receiving a set of keywords T. Assume that the object P_1 (Figure 2.13) is the most textually relevant for the query. Hence, to process this query Q, it is necessary to traverse only the nodes R_7, R_5, and R_1 of the tree shown in Figure 2.14, to reach the object P_1 and return it as query response.

Alternatively, Rocha-Junior et al. (ROCHA-JUNIOR et al., 2011) propose a new structure for indexing spatio-textual data, called Spatial Inverted Index (S2I). This structure optimizes the processing of the top-k Spatial Keyword query. S2I is similar to the Inverted File (ZOBEL; MOFFAT, 2006; ROCHA-JUNIOR et al., 2011), but it stores the most frequent terms of the collection with a different method. S2I maps the most frequent terms of the collection to aggregated R-trees (aR-tree) (PAPADIAS et al., 2001), where each tree stores only objects that have the same term t. In like manner, S2I stores the less frequent terms in file blocks, where each block stores objects that have the same term t.

term	id	pf_t	*flag*		storage structure
scholl	t_1	4	tree	→	aR^{t_1}
nursery	t_2	3	tree	→	aR^{t_2}
childlike	t_3	3	tree	→	aR^{t_3}
language	t_4	1	file block	→	(p_5)

Figure 2.15 *Spatial Inverted Index.*

The S2I (exemplified in Figure 2.15) is composed of vocabulary, file blocks (b_i) and aR-tree's (aR^{t_i}). The vocabulary stores each distinct term in the database (e.g., "school", and "nursery"). For each term t_i, it stores the amount of objects pf_t in which t_i occurs. Also, it stores a flag indicating what type of structure the term is stored in (block or tree), and a pointer to the structure containing the term (represented by the unidirectional arrow in Figure 2.15).

Each block file stores a set of objects. For each object in this set, it stores the object's identification $p.id$, the object location $p.l$ and the frequency $f_{p,t}$ with which the term t occurs in the textual description of the object p.

The leaf nodes of the aR-tree store the same information as the file blocks: $p.id$, $p.l$ e $f_{p,t}$. The intermediate nodes store a Minimal Bounding Rectangle (MBR) that involves the spatial location of all objects that are in the subtree. The intermediate node also stores a non-spatial value, representing the maximum value of $f_{p,t}$ of the objects stored in the subtree (PAPADIAS et al., 2001). Thus, objects can be accessed decreasingly by $f_{p,t}$ values, and spatial proximity (ROCHA-JUNIOR et al., 2011).

According to Rocha-Junior et al. (2011), the results obtained using S2I demonstrate the cost optimization of the query, as well as the cost to update an existing term in the collection (ROCHA-JUNIOR et al., 2011). For queries with only one keyword, S2I traverses only a small tree or file block. When the query has several keywords, it is necessary to go through only a set of small trees or blocks of files, dispensing access to an external inverted index.

2.8 TOP-K SPATIAL KEYWORD PREFERENCE QUERY

Top-k Spatial Keyword Preference Query (SKPQ) is a query proposed by Almeida e Rocha-Junior (2015) similar to the Top-k Spatial Keyword query (SKQ). Likewise the SKQ, a significant part of the traditional spatial queries are user-centered. Most types of query search for spatial objects consider the user position. This is the case of the spatial queries *range* and *nearest neighbor* (*nn*). The range selects objects that are within a distance r (radius) of the user location, while *nn* returns the closest spatial object from the user location. This is also the case of the Top-k Spatial Keyword Query that returns the k most relevant spatio-textual objects by considering both the distance between the spatio-textual objects and the user location, and the relevance between the text of the spatio-textual objects and the query keywords (as discussed in Section 2.7).

Differently from the user centered spatial types of query, the SKPQ query searches for interesting objects considering other spatio-textual objects (features) in their spatial neighborhood. Specifically, given a set of interesting objects (e.g., hotels), a set of features (e.g., bars, restaurants and tourist attractions), a spatial selection criteria (e.g., 100m from the spatial objects of interest) and a set of query keywords (e.g., "Italian food"); the SKPQ returns the k best interesting objects, where the score of each object is given by the highest textual relevance between the query keywords and the text of the features that satisfies the spatial selection criteria.

In other words, the SKPQ is a preference query that uses query keywords to describe the user preference. The SKPQ searches for spatial objects of user's interest based on features in their spatial neighborhood. For example, Figure 2.16 describes a spatial area with spatial objects p (e.g., hotels) and features f (e.g., any establishment). Consider a user interested in book a hotel close to a Japanese restaurant. The user specifies the query keywords "japanese restaurant" and the spatial selection criteria (represented by the circle around the objects p). An evaluation method defines that the textual description of the object f_1 "restaurant" has textual relevance to query keywords. However, the textual description of object f_4 "japanese restaurant" is more textual relevant because it has the same words as the query keywords. Objects f_2, f_3, f_5, f_6, f_7 have no textual relevance to the query keyword, while f_5 does not satisfy the spatial selection criteria too. The SKPQ returns the object p_3 as the best hotel for the user's need, since f_4 has the greatest textual relevance among all features and satisfies the spatial selection criteria.

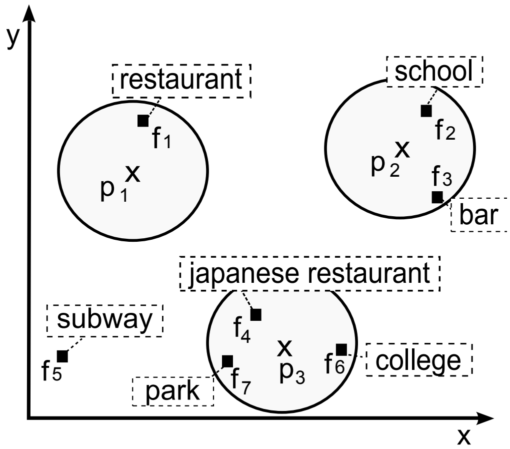


Figure 2.16 Interesting objects (p) and features (f) associated with their textual descriptions

Chapter
3

THE SEMANTIC WEB

In 2001 Tim Berners-Lee stated: "*Most of the Web's content today is designed for humans to read, not for computer programs to manipulate meaningfully*" (BERNERS-LEE et al., 2001). Indeed, web applications can parse a web page for layout and text processing. For example, it is possible to identify a header, or a link, to extract information about the content in the page. However, they have no reliable way to process the semantics. The Semantic Web brings structure to the meaningful content of web pages, enabling web applications to answer sophisticated user queries without use complex artificial intelligence solutions. The Semantic Web is an extension of the World Wide Web that improves data sharing, discovery, integration, and reuse. In order to achieve these goals, the Resource Description Framework (RDF) and the Web Ontology Language OWL is employed. RDF describes knowledge graphs, while OWL expresses type logics (called as *ontologies*) attached to these graphs (SARKER et al., 2017).

Together with these new resource descriptions technologies, arises the need for a new query language to extract the information. Since the RDF release, several query languages have been proposed (see (HUTT, 2005) for further description). In 2004, the RDF Data Access Working Group released the first draft of SPARQL - a query language for RDF. In essence, SPARQL is a graph-matching query language where the query consists of a pattern which is matched against a data source. The values obtained from this matching are processed and generates the answer to the user (PÉREZ; ARENAS; GUTIERREZ, 2009).

The goal of this chapter is to present concepts related to the Semantic Web. The chapter consists of three main sections: i) Section 3.3 introduces Linked Open Data, ii) Section 3.1 presents the basics of RDF, Section 3.4 presents the SPARQL query and illustrates how use it, and Section 3.5 concludes the chapter.

3.1 RESOURCE DESCRIPTION FRAMEWORK - RDF

The Resource Description Framework (RDF) is a framework for representing information in the Web. It has an abstract syntax and formal semantics which enable deductions in RDF data.

RDF represents information in a minimalist and flexible way. RDF usually shares information between applications which have individually design setups. This framework increases the value of information as it becomes accessible to more applications across the entire Internet (KLYNE; CARROLL, 2006).

3.1.1 Graph Data Model

The RDF structure is a collection of triples which each contains a subject, a predicate, and an object. A set of such triples is called an RDF graph. Figure 3.1 illustrates an RDF graph using a node and directed-arc diagram. In this graph, each triple is represented as a node-arc-node link (for this reason the term "graph" is employed) (KLYNE; CARROLL, 2006; PAN, 2009).

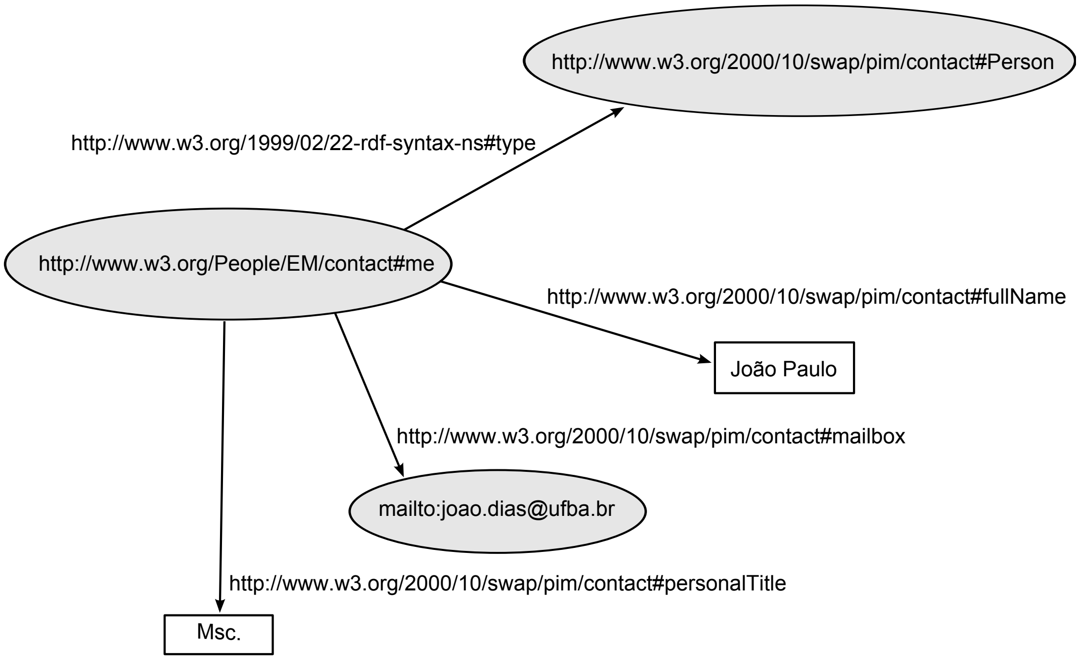

http://www.w3.org/2000/10/swap/pim/contact#Person
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.w3.org/People/EM/contact#me
http://www.w3.org/2000/10/swap/pim/contact#fullName
João Paulo
http://www.w3.org/2000/10/swap/pim/contact#mailbox
mailto:joao.dias@ufba.br
http://www.w3.org/2000/10/swap/pim/contact#personalTitle
Msc.

Figure 3.1 An example of an RDF graph describing a person.

Each triple expresses a statement of a relationship between the nodes that it links. Each triple has three parts:

1. a subject,

2. an object, and

3. a predicate (also known as property) that denotes a relationship.

The nodes of an RDF graph are its subjects and objects, while the arcs are its predicates. The arc always points toward the object. For instance, Figure 3.1 exemplifies an RDF graph describing a Person identified by <http://www.w3.org/People/EM/contact#me> (subject), whose name is João Paulo, whose email address is joao.dias@ufba.br, and whose title is Msc. The predicates are the URIs near the arcs (e.g., <http://www.w3.org/2000/10/swap/pim/contact#mailbox>) and the objects are the values insides the rectangles or ellipses. An object can be a literal (e.g., João Paulo), an RDF URI reference (<http://www.w3.org/2000/10/swap/pim/contact#Person>) or a blank node.

In essence, an RDF triple denotes that some relationship, indicated by the predicate, holds between the things denoted by subject and object of the triple. According to (KLYNE; CARROLL, 2006), the assertion of an RDF graph amounts to asserting all the triples in it. Therefore, the meaning of an RDF graph is the conjunction (logical AND) of the statements corresponding to all the triples it contains.

3.2 ONTOLOGIES

Ontology provides a foundation for the common understanding of some area of interest among people. Even if the people do not know each other, or have different traditions and languages, the ontology may be enough to make them understand each other (DIETZ, 2006). In other words, an ontology is a formal specification of a shared conceptualization (GRUBER, 1995). *Conceptualization* stands for the concept meaning and its relationships in a domain, while *specification* stands for the formal, declarative, and explicit definition of this concept and its relationships.

The Web Ontology Language (OWL) is used in the Semantic Web to describe the relationship between concepts formally. In effect, machines and humans can understand ontologies represented by OWL. Ontologies provide a common concept structure where shareable and reusable LOD datasets are built. Therefore, ontologies facilitate interoperability and data incorporation. In addition, OWL enables applications to make precise inferences like class or instance inferences, without requiring the description of all concepts relationships

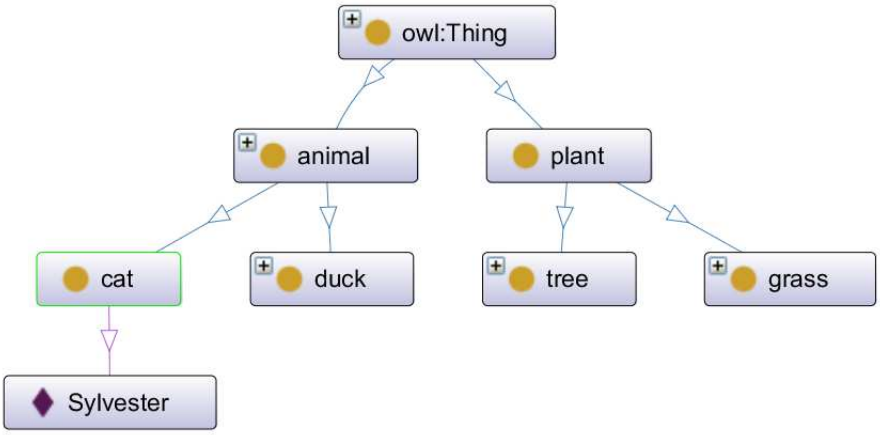


Figure 3.2 An ontology graph representing concepts and relationships between concepts.

Ontology classifies things in terms of semantics or meaning. In OWL, this is achieved through the use of classes, subclasses, and instances (individuals). Figure 3.2 illustrates an ontology graph describing classes and subclasses. Usually, the root node in the ontology graph is *owl:Thing*. In essence, every concept is a subclass of this root node. We can observe that this ontology defines *cat* and *duck* as subclasses of *animal*, and *tree* and *grass* as subclasses of *plant*. The individuals are members of a given OWL class, so we can define "Sylvester" as a member of the cat class. This way, we can infer that "Sylvester" is an animal too because of *cat* is subclass of *animal* in the ontology graph.

3.3 LINKED OPEN DATA - LOD

The Web has evolved into a space where both documents and data are linked (BIZER; HEATH; BERNERS-LEE, 2009). In order to support this new Web, a set of practices for publishing and connect structured data has been proposed by Berners-Lee (BERNERS-LEE, 2011). This set of practices is known as Linked Data because it enables a user to start browsing in one data source and then navigate along with links into related data sources. In addition, Linked Data is published in such a way that the data is machine-readable, enabling new possibilities for applications. Berners-Lee (BERNERS-LEE, 2011) defines the following set of practices to create Linked Data:

1. Use Uniform Resource Identifiers (URIs) as name for things;
2. Use HTTP URIs to publish your data;
3. Provide useful information using the standards (RDF, SPARQL);
4. Include links to other URIs, so users can discover more things.

In a nutshell, Linked Data relies on these three technologies: Uniform Resource Identifiers (URIs) (BERNERS-LEE; FIELDING; MASINTER, 2005), the HyperText Transfer Protocol (HTTP) (FIELDING et al., 1999), and the Resource Description Framework (RDF) model. A simple way to create linked data is using one RDF file with a URI which points into another file. Suppose an RDF file, named <http://example.org/Hotels>, where hotels around the world are described. Local identifiers (Venice, Italy and Hotel_Danieli) are used to describe one hotel (resource). In Listing 3.1, hotel Danieli is described with RDF. An HTTP URI <http://example.org/Hotels/Hotel_Danieli> can be assigned, enabling anyone on the Web to access the hotel's description.

```
<rdf:Description about="Hotel_Danieli"
  <rdf:type rdf:Resource="Italy">
  <rdf:type rdf:Resource="Venice">
</rdf:Description>
```

Listing 3.1 Description of hotel Danieli in an RDF file.

Now, suppose there is another RDF file (listing 3.2) containing the description of Hotels in Venice. Hotel Danieli is in Venice; however, there is no need to describe it again in Listing 3.2). Hotel Danieli is described by its HTTP URI which points to its description. When these files are released under an open license, they are called Linked Open Data (LOD). In this work, we use two LOD sources: DBpedia and LinkedGeoData. These datasets are described in Section 5.1.

```
<rdf:Description about="Hotels_in_Venice"
  <rdf:type rdf:Resource="http://example.org/Hotels/Hotel_Danieli">
</rdf:Description>
```

Listing 3.2 Description of hotels in Venice in a RDF file.

Tim Berners-Lee Berners-Lee (2011) suggested a 5-star rating system for Linked Open Data. The more stars the data has, the more shareability "power" it contains. Below, we describe what is necessary to achieve each star:

- **1 star** - Data available on the Web under an open license. Even a PDF or image scan is allowed whether the information is public.
- **2 star** - Data delivered as structured (machine-readable) data. For example, a Excel file instead of a image scan of table.
- **3 star** - Data available in a non-proprietary open format like using CSV instead of Excel.
- **4 star** - All requirements above plus using open standards from W3C (e.g., RDF and SPARQL) to identify things and properties. Following this standard, users can point their data at other data.
- **5 star** - All requirements above plus link your data to other data to provide context.

A notable example of LOD usage is the *Linked Open Data (LOD) Project* that started in 2007 to offer public access to LOD datasets. In 2019, this project connected 1,239 datasets with 16,147 links between them (MCCRAE, 2019), resulting in more than 31 billions items (FRESSATO, 2019). This collection of datasets is known as the LOD cloud. As a result, web search engines can use HTTP URIs to access data within different LOD datasets, effortlessly generating new (and possibly more precise) information. Moreover, applications obtain other benefits from LOD, such as facilitate data reutilization, extension, and shareability (TRIPERINA et al., 2015).

The central node in the LOD cloud is the DBpedia dataset. It has derived its data corpus from Wikipedia, a heavily visited and under constant revision online encyclopedia. The DBpedia Association maintains the dataset and provides an HTTP service endpoint to execute queries. To query data in a LOD dataset one must submit a query using SPARQL language. For this reason, the endpoint usually is called SPARQL endpoint. One can ask queries against DBpedia using the OpenLink Interactive SPARQL Query Builder (iSPARQL)[1], the SNORQL query explorer [2], or any other SPARQL-aware client(s). In this research, we use ARQ[3] to access DBpedia. ARQ is a SPARQL processor for Jena - a free open source framework for building Semantic Web and Linked Data applications.

[1] http://dbpedia.org/isparql

[2] http://dbpedia.org/snorql

[3] https://jena.apache.org/documentation/query/service.html

3.4 SPARQL

SPARQL is a query language that can be used to express queries across diverse data sources. The data queried using SPARQL might be stored natively as RDF or viewed as RDF via middleware. A SPARQL endpoint is used to enable users to query a knowledge base via the SPARQL query language. DBpedia and LinkedGeoData endpoints can be accessed at http://dbpedia.org/snorql/ and http://linkedgeodata.org/sparql. Listing 4.1 introduces a SPARQL query to obtain features within 200 m from a point of interest. In Listing 4.1, *objectURI* is a URI to a point of interest.

SPARQL contains capabilities for querying graph patterns along with their conjunctions and disjunctions. Essentially, a SPARQL query consists of a pattern which is matched against a data source, and the values obtained from this matching are processed to give the answer. The results of SPARQL queries can be result sets or RDF graphs. Listing 3.3 introduces a SPARQL query to obtain objects within 20 km radius of New York City.

```
PREFIX dbr: <http://dbpedia.org/resource/>
SELECT DISTINCT ?resource ?label ?location
WHERE {
    dbr:New_York_City geo:geometry ?sourcegeo.
    ?resource geo:geometry ?location;
    rdfs:label ?label.
    FILTER( bif:st_intersects( ?location, ?sourcegeo, 20 )).
  }
```

Listing 3.3 SPARQL query to obtain objects within 20 km radius of New York city.

The predicate *geo:geometry* is defined at Geo-SPARQL (PERRY; HERRING, 2012), an ontology that represents features and geometries. In Listing 4.1, the variable *location* matches with the spatial coordinates of objects around a point of interest. The function *bif:st_intersects()* returns true if there is at least one point in common between the spatial coordinates *location* and *sourcegeo*. The tolerance for the matching in units of linear distance is supplied at the third parameter of *bif:st_intersects()*. The tolerance is 200 m as illustrated at Listing 4.1.

3.5 SUMMARY

In this chapter, we presented an overview of the Semantic Web. We started by introducing the Resource Description Framework. We also presented Linked Open Data as one of the core concepts of this new Web. Then, we introduced SPARQL and presented how to use it to find spatial objects in a search space.

Chapter

4

PRELIMINARY PROPOSAL TO IMPROVE QUERY RESULTS

The goal of this chapter is to introduce the process of query improvement based on feature textual description enhancement and results personalization. Figure 4.1 describes an overview of our approach where the query result is improved automatically. Given a user query, the search engine first searches for features that satisfy the spatial selection criteria (i.e., inside the defined range). Then, it enhances the feature's textual description using LinkedGeodata and DBpedia. A score function defines the relevance of each feature description based on the query keywords. Subsequently, the objects are ordered according to their scores and sent to the personalization algorithm. Finally, the personalization algorithm re-order the objects based on the user reviews and send the results to the user.

Thus, this chapter presents the details on how our approach improves the Spatial Keyword Preference query. This chapter consists of the following sections: i) Section 4.1 presents the SKPQ enchancement with LOD and the query personalization, ii) Section 4.2 compares the related work with the proposal, and iii) Section 4.3 concludes the chapter.

4.1 PROPOSED ALGORITHM

In this section, we present the proposed algorithms to process the Spatial Keyword Preference query with LOD (SKPQ-LD) and to personalize the query results. The algorithm that process SKPQ-LD employs SPARQL to obtain the textual description for features. The traditional SKPQ uses a Spatial Inverted Index (S2I) to index a text file with all textual descriptions needed. The S2I structure is explained in Section 2.7.1.

Once an object f is found in spatial vicinity of one object p ($dist(p, f) \leq r$), its abstract is accessed using SPARQL. This abstract obtained from DBpedia is the textual description $f.D$. The textual score of f is computed using the cosine similarity function between the query keywords and the abstract $f.D$. Finally, the textual score of f is attributed to p if the score of f is higher than the current score of p ($p.score$), ($\theta(f.D, Q.D) > p.score$). The k objects with the highest scores are maintained in a heap

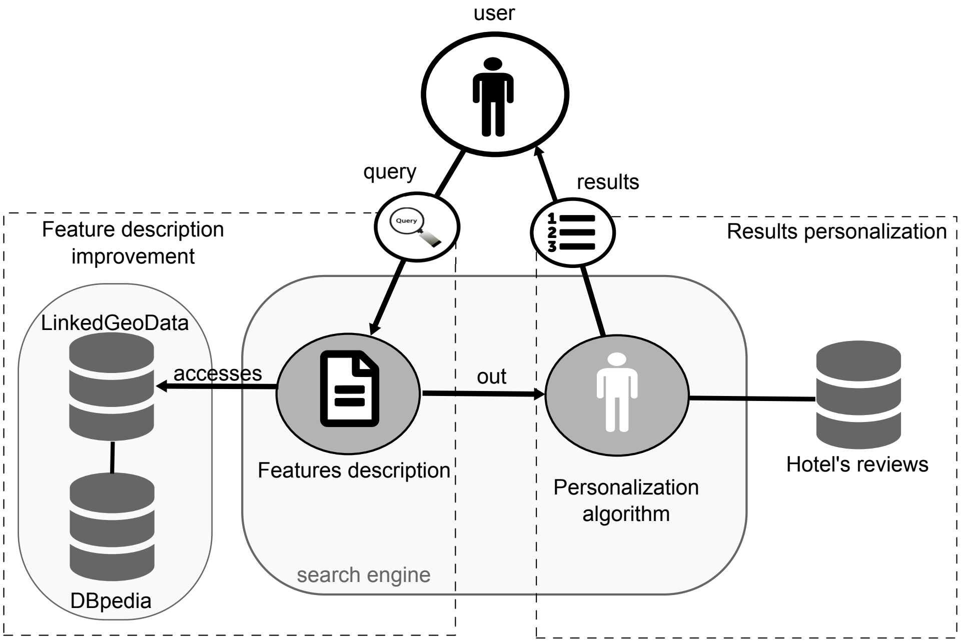


Figure 4.1 Overview of our approach to automatically improve query results.

H of k size. After computing the score of all objects p, the heap contains the k best points of interest. Summarizing, the first step to process SKPQ requires finding all features in the spatial vicinity of each point of interest (POI). Then, compute the textual relevance of each feature (f) to the query keywords, in order to compute the score of each object p. These steps are repeated for all points of interest to obtain the k best ones.

In traditional SKPQ, the textual description of a feature is obtained from S2I. Given a spatial location and one term (keyword), the S2I returns one list with all features that satisfy both the textual relevance and spatial selection criteria. In this proposal, we query data from the LOD cloud with two objectives: 1) to find features that satisfy the spatial selection criteria, and 2) to obtain their textual description. Listing 4.1 describes the SPARQL query used to achieve the first objective. While the SPARQL query used to accomplish the second objective is described in Listing 4.2. More specifically, we access the DBpedia endpoint to read the abstract and comment properties values of each feature obtained. *referenceObjectURI* in Listing 4.2 is the URI to a feature.

```
SELECT DISTINCT ?resource WHERE {
        ?objectURI geo:geometry ?sourcegeo.
        ?resource geo:geometry ?location ;
        rdfs:label ?label .
FILTER( bif:st_intersects( ?location, ?sourcegeo, 0.2 ) ) . }
```

Listing 4.1 SPARQL query to find features that satisfies the spatial selection criteria.

Algorithm 1: Processing SKPQ-LD.

```
   Input: $Q = (Q.D, Q.r, Q.k)$
   Output: Heap that maintains the $k$ best points of interest.
 1 $H \leftarrow \emptyset$ //Heap that maintains the $k$ best points of interest.
 2 for each $p \in P$ do
 3     $p.score \leftarrow 0$
 4     $iterator \leftarrow findObjectF(objectP).iterator()$
 5     while $iterator.hasNext()$ do
 6         $text \leftarrow getAbstract(iterator.next())$
 7         $f.\theta \leftarrow cosineSimilarity(text, Q.D)$
 8         $updateScore(p, f.\theta)$
 9     end
10     if $|H| < k$ OR $p.score > H.peekMin().score$ then
11         $H.add(p)$
12         if $|H| > k$ then
13             $H.removeMin()$
14         end
15     end
16 end
17 return $H$
```

```
SELECT DISTINCT * WHERE {
        ?referenceObjectURI dbo:abstract ?abstract;
        rdfs:comment ?comment.
FILTER( lang( ?abstract)="en"&&lang(?comment)="en") }
```

Listing 4.2 SPARQL query to obtain textual description for one feature.

After the best k objects are found and ordered according to their respective scores, the personalization algorithm re-orders the rank. This new rank reflects the user personal preference, placing the POI that satisfies the query keywords and her preference in the top of the rank.

4.1.1 Feature Description Enhancement Algorithm

In traditional SKPQ, the textual description of features is previously indexed using S2I. The indexing process has a high computational cost but enables the query processing in an optimized way. Instead of computing the textual score of every feature that satisfies the spatial selection criteria (lines 5-9 of Algorithm 1), the S2I provides an iterator that accesses only the features with textual relevance and that satisfy the spatial selection criteria. The S2I avoids the score calculation of features that are in the spatial vicinity of a point of interest but has no textual relevance to the query keywords.

Algorithm 1 presents the algorithm to process the SKPQ-LD. It receives as input the SKPQ $Q = \{Q.D, Q.r, Q.k\}$, where $Q.D$ is the query keywords, $Q.r$ is the radius that defines the spatial selection criteria, and $Q.k$ is the number of expected results. The algorithm computes the score of each object $p \in P$ (lines 2-17). Initially, the score of p is zero (line 3). Then, an iterator (line 4) is employed to access all features f in the spatial vicinity of p. The textual description of each feature f is accessed (line 6), and the textual relevance between this description and the query keywords is computed (line 7) using cosine similarity. In this work, we use cosine similarity because we want the term frequency to be determinant over the document length (ZOBEL; MOFFAT, 2006). The $getAbstract(iterator.next())$ (line 6) process the SPARQL query described in Listing 4.2 to obtain the objects' textual description. After computing the score of the feature f, the function $updateScore(p, f.\theta)$ updates the score of p if the feature's textual score $f.\theta$ is higher than the current score of p (line 8).

An object p is added into H only if H has less than k objects or if the score of p is higher than the lowest score among the objects currently stored in H ($p.score > H.peekMin().score$). If the size of H is larger than k, the object with the smallest score in H is removed (lines 10-15). The algorithm returns the k objects p with the highest scores stored in H (line 17).

As shown above, the algorithm to process the SKPQ-LD computes the score of each object $p \in P$ calculating the textual relevance between $Q.D$ and each $f' \in F'$, where F' is a subset of F ($F' \subseteq F$) that contains the feature f' that satisfies the spatial selection criteria. Hence, the complexity of the algorithm is $O(|P| \cdot |F'|)$.

4.1.2 Query Result Personalization Algorithm

After the SKPQ-LD is processed and the heap H with the results are generated, the personalization algorithm (Algorithm 2) updates the score of each POI in H. For each $p \in H$, a set of reviews R describing the POI p is obtained from a reviews database (line 2). A classifier, trained with query user reviews to any POI, classifies each object review $r \in R$ as good or bad to the query user. In fact, the classifier compares the query user review with the ones in the reviews database. Whether the query user review is similar to r, it receives a value of 1 (good); otherwise, it receives 0 (bad).

Each set of reviews R contains a different number of reviews describing the POI. For this reason, we employ an accumulator c to increase when the review is classified as good or to decrease when the review is classified as bad (lines 5-10). Then, the accumulator value is normalized as described in line 11. In the end, the POI score is updated (line 12), changing the p rank position according to the user preference described in the user profile.

4.2 RELATED WORK

Several studies employ LOD datasets to improve textual descriptions of spatial objects. Hegde et al. (2011) describe an augmented reality browser that uses LOD to enhance the description of objects, offering a better recommendation. The objects were represented by a semantic relationship between them and several spatial data repositories such as

Algorithm 2: Personalization algorithm.

```
Input: $H_k = \{p_1, p_2, ..., p_k\}$
Output: $H_k$
for each $p \in H$ do
    $R \leftarrow getReviewSet(p)$
    $c \leftarrow 0$
    for each $r \in R$ do
        if $classify(r) == 1$ then
            $c++$
        end
        else
            $c--$
        end
        $c \leftarrow \frac{c}{|R|}$
        $p.score \leftarrow p.score + c$
        $H.update(p)$
    end
end
return $H$
```

Wikipedia[1] and YouTube. Using Natural Language Processing techniques, the user's profile is semantically related to a point of interest (POI). Then the personalized set of POIs is delivered to the user. Similarly, Karam e Melchiori (2013) present a way to improve POIs description using LOD. They developed the M-PREGeD, a conceptual framework aiming to improve the accuracy of spatial data from different LOD sources. In M-PREGeD, voluntary users can generate or update POIs descriptions in order to enhance it. Aiming the same goal, we use DBpedia to enhance the textual description of features. However, we do not make use of voluntary users to help the process because we aim for an automatic enhancement approach.

The popularization of GPS (Global Positioning System) enabled devices increases significantly the volume of spatial data produced in the last years. This phenomenon stimulates new systems making use of spatial data associated with LOD. Fernández-Tobías et al. (2011) use LOD and spatial data to recommend musicians related to the architecture around the user's spatial position. Likewise our approach, they used LOD to obtain data about a spatial area (ex: architecture in Rome) but they did not make use of any spatial information (ex: latitude or longitude) in their recommendation. We use spatial information to select objects that satisfy the user's information need. Equally important, Becker e Bizer (2009) present a location-aware semantic web client for mobile devices, named DBpedia Mobile. The web client uses the current GPS position to render a map where the user can explore information about his surroundings with linked data. This

[1] https://www.wikipedia.org/

information is obtained by navigating along with data links into other data repositories. In our work, we use the semantic representation of spatial objects available at DBpedia to measure the similarities between the user's keywords and the feature.

Accordingly Braun, Scherp e Staab (2010), simple text description hinders the extraction of relations between objects. In order to mitigate this problem, they propose a semantic representation of objects using LOD. They created a collaborative spatial database compounded by POIs. In this database, users can define the ontology category of each POI. To improve the POI quality, a revision engine based on data mining techniques is provided. The revision engine identifies duplicate POIs with similar annotations or slightly varying locations for the same spatial location. In a similar fashion, Nikolaou et al. (2013) present a tool to explore linked spatial data as well as create and collaboratively edit thematic maps. Despite this tool does not provide a revision engine to improve POI quality, it provides an exploration of linked spatial data that span across multiple SPARQL endpoints. By querying these endpoints, the user is able to create his own maps and share these maps with others. The linked spatial data is explored by a tool that builds a class hierarchy and discovers the spatial extent of available information. In our work, we do not use GeoSPARQL (BATTLE; KOLAS, 2011) but we use a SPARQL extension to explore the spatial vicinity of each point of interest.

Meta-Knowledge is another approach employed to enrich the textual description. Meta-Knowledge refers to include metadata at textual corpus using an annotation scheme. For example, a news text about an event can include metadata like the modality, subjectivity, source, polarity, and specificity of the event (THOMPSON et al., 2017). This approach enriches the metadata instead of the data describing the object. In this work, we aim to enrich the data describing a spatial object. In a like manner, query expansion has long been suggested for dealing with the issue of word mismatch in information retrieval. Accordingly to Xu e Croft (2017), there is a number of query expansion approaches. The main approaches are to analyze the query description to discover word relationship (global techniques) and analyze the objects retrieved by the query localization (local feedback).

With this in mind, Karpathiotaki et al. (2014) introduce the Prod-Trees platform, a semantically enabled search engine for earth observation products (ex: products derived from aerial or satellite imagery). The platform has a web interface that allows users to submit free text queries. A query analyzer uses Linked Data to display different interpretations for the inserted query. The user selects the interpretation she wants, then the backend service generates queries and sends them to a catalog service. When the catalog service is ready, the results are sent to the user. In our approach, the user is able to submit free text queries as well as in the Prod-Trees platform, but we do not use a query analyzer to expand the query.

Location-based services (LBS) are aggregating relevant information about users, their behavior and their preferences based on the location histories. Personalizing these services offer to users relevant information tailored to their preferences and behavior. Recent, some approaches to deal with personalization of query results have been proposed. Kwon e Shin (2008) propose a personalized location-aware query methodology based on the user's current location and schedule. They apply a contextual concept distance to asses if a user is interested in visiting a POI at the moment she executes the search. The user's schedule is used to personalize the query results, presenting only POIs that are relevant to the user's scheduled activities. Similarly, we personalize the query results using users reviews to give more relevance to objects similar to those the user enjoyed in the past.

Guo, Alamudun e Hammond (2016) describe the RésuMatcher - a personalized system to find jobs. The user submits his résumé in the system and it presents jobs descriptions which contain skills similar to those described in the user's résumé. An ontology-based similarity measure is employed to compare the skills in the résumé with the skills in job descriptions. While they use only an explicit interaction (résumé) to personalize their system, our solution uses both implicit (users reviews) and explicit (keywords) interactions to achieve the same goal.

Urban freight management is another task that requires personalization solutions. Aiming at this problem, Bouhana et al. (2015) present an information retrieval method for a personalized itinerary search in urban freight transport systems. In this system, the user submits a request containing several topics to describe her preferences. Then, they combine a Case Base Reasoning and SWRL rules to personalize the query results. Solving another routing problem, Dai et al. (2016) propose the Personalized and Sequenced Route (PSR) Query. The authors enabled the user to define weights to POIs in order to obtain a personalized route between two spatial locations that pass by the preferred locations. Hence, the PSR query considers multiple factors of a route and different weights distributed by the user on all objects of his interest. According to Kwon e Shin (2008) and Allan et al. (2012), a location-aware service should ask for the minimum user input possible. Therefore, our approach uses just the query keywords to describe the user need and user reviews to personalize the query result.

4.3 SUMMARY

This chapter presented the preliminary proposal to improve the Spatial Keyword Preference Query results. This approach employs LOD to increase the number of words describing a spatial object and reorder the rank using query personalization. In addition, this chapter presents the proposal in details along with the algorithms to process the improvement. Finally, it describes the related work, comparing them with the proposal.

Chapter 5

PRELIMINARY PROPOSAL EVALUATION

In this chapter, we present our methodologies and the results obtained during the preliminary proposal evaluation. In addition, we discuss the dataset and the methodologies employed to analyze the proposed algorithm. The experiments to evaluate the feature description enhancement were performed in two ways, each with a unique methodology. In the first experiment (Experiment 1), the users' ratings from Google Maps were extracted to evaluate the queries result. In the second experiment (Experiment 2), the users' ratings were extracted from TripAdvisor[1].

5.1 DATASETS

In this work, we used three datasets to process the SKPQ. The OpenStreetMap[2] dataset was used to process SKPQ and, DBpedia and LinkedGeoData were used to process SKPQ-LD. Additionally, two publicly available datasets were used to evaluate the obtained query results: the Google Maps dataset and the OpinRank dataset.

Extracts are pieces of OpenStreetMap data pruned at the region of individual continents, countries, or metropolitan areas. Mapzen[3] maintains updated extracts for many cities. In this work, we used Mapzen to obtain OpenStreetMap data from Dubai. We process this dataset to extract only spatio-textual objects. The set of points of interest P is composed by spatial objects whose the category in the OpenStreetMap is hotel, while the set of features F is composed by the other spatio-textual objects. Table 5.1 presents some characteristics of the dataset obtained at Mapzen: the number of points of interest $|P|$, the number of features $|F|$, the number of unique terms in the dataset and the total number of terms.

LinkedGeoData uses the information collected by the OpenStreetMap project and makes it available as an RDF knowledge base according to the Linked Data principles.

[1] https://www.tripadvisor.com.br/

[2] http://www.osm.org

[3] https://mapzen.com/data/metro-extracts/

Dataset	$\lvert O \rvert$	$\lvert F \rvert$	No. of unique terms	Total number of terms
Dubai	162	2243	1906	12256

Table 5.1 Characteristics of the data obtained at Mapzen.

To process SKPQ-LD we used SPARQL at LinkedGeoData to obtain a set of objects P equivalent to the one obtained from Mapzen, as illustrated by Listing 5.1. This SPARQL query returns a list of objects with the same name as the one stored at Mapzen, but different spatial coordinates (i.e., there are several places called “McDonald’s” in Dubai, but at different spatial coordinates). Then, we selected only the object with the same name and the same spatial coordinate as the one selected as p object at Mapzen. Additionally, we used the LinkedGeoData endpoint to access the feature’s textual description. The textual description obtained from LinkedGeoData is composed by *rdf:type* and *rdfs:label* predicates.

```
SELECT * WHERE {
        ?var rdfs:label "OSMlabel" .
        ?var geo:lat ?lat.
        ?var geo:long ?lon. }
```

Listing 5.1 SPARQL query to obtain the points of interest to process SKPQ-LD.

In order to enrich the object’s textual description from LinkedGeoData, we used the data obtained from DBpedia. The DBpedia project has derived its data corpus from the Wikipedia encyclopedia, a large collaborative encyclopedia. When a feature has the same *rdfs:label* in DBpedia and in LinkedGeoData, we concatenate the text obtained in both datasets. The textual description $f.D$ obtained from DBpedia is composed by *rdfs:comment* and *dbo:abstract* predicates. As an example, the Hotel Danieli from Venice is described as “(tourism) (hotel) Danieli” in OpenStreetMap. While in DBpedia, the same hotel is described as “Hotel Danieli, formerly Palazzo Dandolo, is a five-star palatial hotel in Venice, Italy. (..)”[4]. The hotel description in DBpedia is much wider than the OpenStreetMap description, with 58 more words.

Both DBpedia and LinkedGeoData have public access. We accessed the data from their respective endpoints, storing the obtained data in a local repository. When the query searches for the textual description of one object, it first searches in the local repository. If the search fails, it looks for the information in the endpoints.

5.1.1 Dataset for Experiment 1

Besides the datasets used to process the SKPQ and SKPQ-LD, we used the Google Maps dataset and OpinRank dataset to evaluate the queries. The Google Maps dataset was accessed through the Google Places API. This dataset contains points of interest that are updated frequently through owner-verified listings and user-moderated contributions. We extract from Google Maps the users’ ratings to the hotels retrieved by the SKPQ and SKPQ-LD. These users’ ratings are used to evaluate both SKPQ and SKPQ-LD.

[4] Full description can be accessed at http://dbpedia.org/page/Hotel_Danieli

Hotel name	Aspect Rating Value
Hatta Fort Hotel	4.107
Al Manzil Hotel	4.341
Park Hyatt	4.342

Table 5.2 Example of information available in OpinRank dataset related to the query "great location".

5.1.2 Dataset for Experiment 2

The OpinRank dataset (GANESAN; ZHAI, 2011) contains hotel reviews and aspect ratings. There are 5 aspects ratings related to hotels: *cleanliness*, *value*, *service*, *location* and *room*. The aspect ratings values are on a scale of 1-5. Ganesan and Zhai (GANESAN; ZHAI, 2011) manually created textual queries related to each aspect rating. These queries were based on real queries made by users in popular search engines, so they reflect a natural user query. For example, the query "great location" is related to the aspect rating *location*. Given the query, the dataset lists the aspect rating value of each hotel as described in Table 5.2. The rating values are given by users from TripAdvisor when evaluating the hotels they have visited. In essence, the OpinRank dataset contains five hotels aspects, each aspect is related to five user queries and one aspect rating value for each hotel as described in Table 5.2.

5.2 METHODOLOGY

The DBpedia and LinkedGeoData were accessed through the local repository, or by the Snorql endpoint, as explained in Subsection 5.1. All experiments were executed in the same computer with an Intel Processor of 1.8 GHz (model i3-3217U) and 8 GB of RAM memory. For processing the SKPQ we made use of OpenStreetMap dataset, while for SKPQ-LD we used DBpedia dataset merged with OpenStreetMap dataset using SPARQL queries as discussed in Section 4.1.

Experiments 1 and 2 employ a specific methodology to evaluate the SKPQ-LD: the first uses ratings obtained from Google Places API, and the second uses relevance judgments obtained from TripAdvisor. In Experiment 1, SKPQ and SKPQ-LD were executed twenty times using one unique query keyword each time. Half of the keywords are the most frequent terms in the dataset, the other half were randomly obtained. The query results were evaluated using NDCG. The list of frequent terms was obtained from S2I[5] and random queries keywords were obtained without repetition from a set of 1906 terms extracted from the OpenStreetMap dataset. "chili" and "sunset" are examples of random keywords used in this work. We used the object rate obtained from Google Places API to determine the ideal ranking.

In Experiment 2, SKPQ and SKPQ-LD were executed using query keywords described in the OpinRank dataset. This dataset contains full reviews of hotels collected from

[5] Implementation available at XXL Library

Tripadvisor and their corresponding aspect ratings as described in Subsection 5.1. We use the queries related to each aspect as query keywords and evaluate the query result obtained by SKPQ and SKPQ-LD. We ordered the query result by the aspect rating value of each hotel to determine the ideal ranking.

The same set of POIs was used to process the traditional SKPQ and the SKPQ-LD. Both queries have the same definition and use cosine similarity to evaluate the textual relevance between query keywords and the feature description $f.D$. We compare the score of each POI computed by these two queries to understand how LOD affects the object evaluation.

In order to evaluate the query personalization approach, we built user profiles and choose the suitable classifier to work with these profiles. The next subsections detail these user profiles and the methodology employed to choose the classifiers.

5.2.1 User Profiles

User profiles are used to describe the user preference, using her past reviews to indicate the best item for her today. As described in Section 5.1, each hotel has five aspect ratings in the Opinrank dataset: cleanliness, room, service, location, and value. We build one user profile for each aspect rating. A user profile consists of twenty user reviews and their respective label (0 for a bad review, 1 for a good review). For instance, when building the *service* aspect rating related profile, a specialist selected ten reviews about the hotels with the highest *service* aspect rating value and another ten about the hotels with the lowest. This way, the user profile represents a user who visited hotels with good and bad services and commented about them on TripAdvisor.

Table 5.3 illustrates an user profile, where each line displays the label followed by the user review. Usually, reviews has a large number of characters (see Appendix A).

Review Label	Review Text
0	Poor Customer Service :(All was good at the early stages, however I was disappointed when I went to the room (...)
1	FABULOUS! We have just returned from a wonderful 8 days at the Residence & Spa.

Table 5.3 Example of the user profile related to service aspect rating

The review text is filtered using the StringToWordVector method (ADELEKE et al., 2018; VARUDHARAJULU; MA, 2019) which converts the string into a vector containing a set of attributes representing word frequency, in other words, representing information from the text contained in the strings. The vectorized profile is used to train a classifier which will identify the good reviews from hotels the user have not visited yet, as described in Section 4.1.2.

5.3 METRICS

The metrics employed in all experiments were Discount Cumulative Gain (DCG), Normalized Discount Cumulative Gain (NDCG) and Mean Average Precision (MAP). These metrics are also used in the referred related works (SONG et al., 2016; SEO et al., 2018; WANG et al., 2015). Higher values indicate better performance under these metrics.

The NDCG is widely used in Information Retrieval, measuring the quality of the ranking produced by a system (BALTRUNAS; MAKCINSKAS; RICCI, 2010; JÄRVELIN; KEKÄLÄINEN, 2002). It is particularly suitable for search applications since it accounts for multilevel relevance. The NDCG corresponds to the value of DCG divided by IDCG, defined in Equation 5.3. Since the top-k items are presented in a rank, then the Discounted Cumulative Gain (DCG) and ideal DCG (IDCG) are calculated based on Equation 5.1 and 5.2, respectively. We denote top-k items by $P_k = \{p_1, p_2, ..., p_k\}$, where the items are ranked by the SKPQ and SKPQ-LD; and we denote rel_i as the relevance value of the item at position i. DCG@k is defined as

$$DCG@k = \sum_{i=1}^{|O_k|} \frac{rel_i}{log_2(i+1)} \tag{5.1}$$

The IDCG is the maximum value of DCG. It is calculated as

$$IDCG = max(DCG@k) \tag{5.2}$$

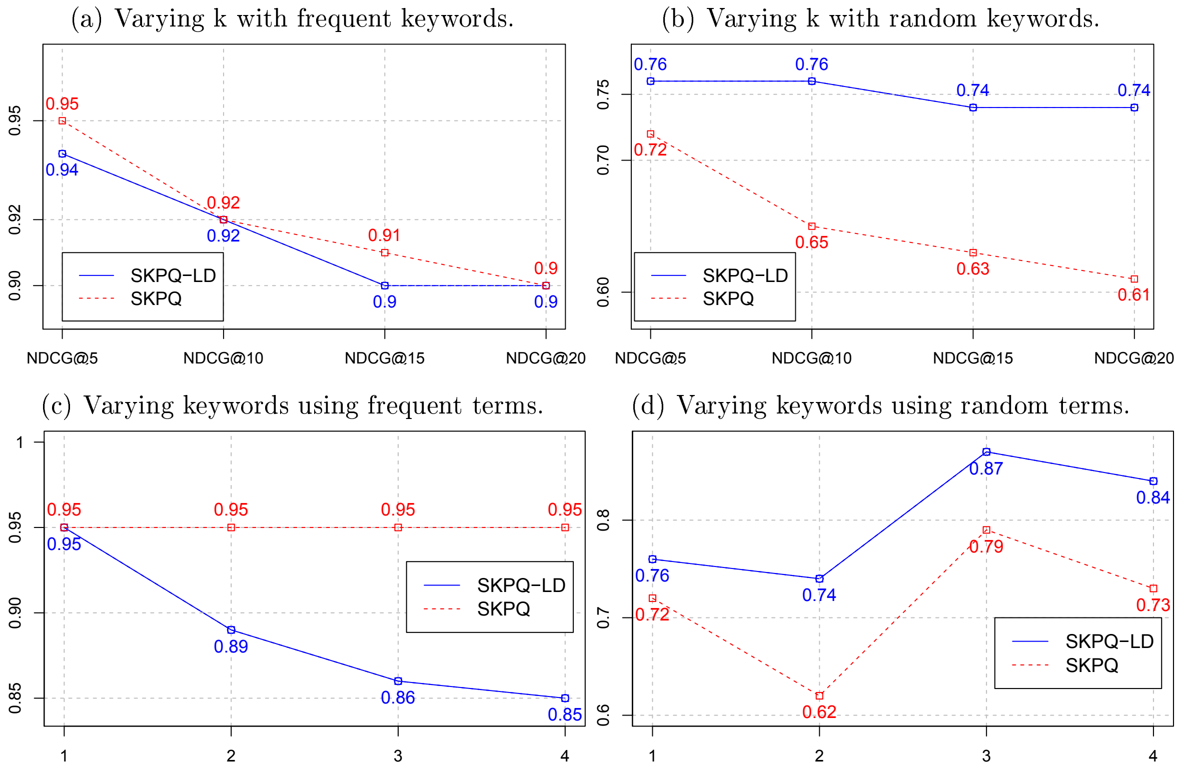
NDCG@k is calculated as

$$NDCG@k = \frac{DCG@k}{IDCG} \tag{5.3}$$

5.4 EXPERIMENT 1: EVALUATING QUERY RESULTS

To understand the ranking quality of both SKPQ and SKPQ-LD, we compared the NDCG values obtained when using random keywords and frequent keywords. Figure 5.1 reports the arithmetic mean of NDCG@k (k=5, 10, 15, 20) that are generated by the queries with different keywords. The arithmetic mean values are reported on the vertical axis. Figures 5.0(a) and 5.0(b) illustrate that SKPQ-LD improves the ranking quality when using random keywords, otherwise the quality is roughly the same.

It is noticeable that we obtain better results with SKPQ using frequent keywords. Since the keyword is present in many objects, there is no problem to SKPQ identify the object that has textual relevance to the query keyword. In this scenario, the objects in SKPQ have a small textual description, but they have a high probability to match with the query keyword. In addition, the SKPQ access more objects because OpenStreetMap offers a larger dataset. Therefore, SKPQ counts on a good enough textual description, and a larger amount of objects, factors that lead to a better evaluation result. Nevertheless, the SKPQ-LD obtained results nearly as good as SKPQ, with a difference of only 0.1 between the NDCG values.


(a) Varying k with frequent keywords.

(b) Varying k with random keywords.

(c) Varying keywords using frequent terms.

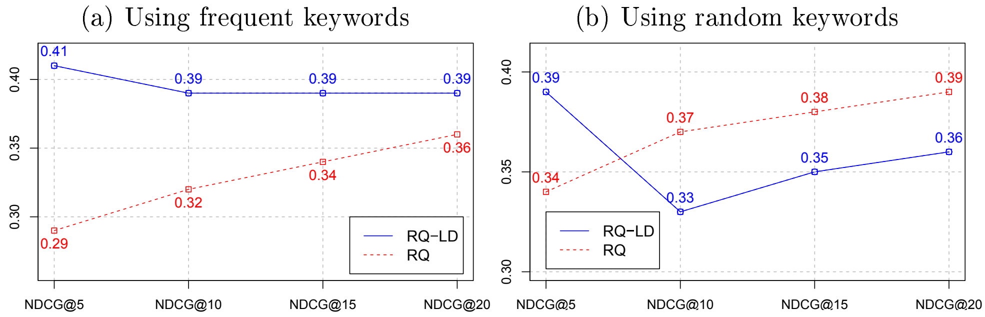
(d) Varying keywords using random terms.

Figure 5.1 Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size (k).

Figures 5.0(c) and 5.0(d) illustrate the NDCG values obtained when varying the number of query keywords. The results depicted in this Figure use a fixed k value of 5. The experiment illustrated in Figure 5.0(c) used the 10 most frequent terms in the dataset as query keywords. To build query keywords with 2 terms or more, we combined these terms with each other without repetition.

As it can be seen in Figure 5.0(c), even after adding three more keywords, the results obtained in SKPQ does not change. On the other hand, SKPQ-LD is more influenced by the increase in the number of query keywords. As observed in Figure 5.1, the SKPQ presents better outcomes with frequent keywords while SKPQ-LD is better with random keywords. However, the distance between NDCG values obtained by SKPQ-LD in Figure 5.0(c) slowly decreases as the number of keywords grows. In addition, we noticed that the SKPQ results had few, or none, changes when the number of keywords was increased. For example, the query result for the keywords "parking cafe" was equal to the query results obtained with "bank parking cafe" and "parking supermarket cafe bank". The textual score of each object presented had changed, but there was no difference on the rank order, resulting in similar NDCG values. The SKPQ lacks a result variability because of the poor textual description of its objects. SKPQ-LD obtained lower NDCG values but did present different results to each query keyword.

As a baseline, the SKPQ query results are compared against the top-k Range Query (RQ) (CAO et al., 2012) results. We employ our approach to enrich the textual description of objects accessed by RQ and evaluate the results obtained. Given a spatial area and the query keyword, the RQ returns k objects in the given area that are textual relevant to the query keyword. All RQ used the same query keywords as SKPQ and a random query location in Dubai. The radius of 200 m from the selected query location defines the spatial neighborhood.


(a) Using frequent keywords

(b) Using random keywords

Figure 5.2 Results obtained with RQ and RQ-LD.

It can be seen in Figure 5.2 that our approach improved RQ result set when using frequent keywords instead of random keywords. The RQ looks for all k objects in a small spatial area (radius = 200 m) while SKPQ looks for objects in the neighborhood of many points of interest. Each object neighborhood has the same size of all the spatial area visited by RQ (200 m). This contrast results in a more challenging effort to build a quality rank for the given area because there are fewer objects to verify. This can be verified observing the much lower NDCG values obtained with RQ. While SKPQ obtained 0.61 in its worst case, RQ obtained 0.41 as its best case. The amount of objects to verify is the main reason for the lower NDCGs values depicted in Figure 5.2 than the ones in Figure 5.1.

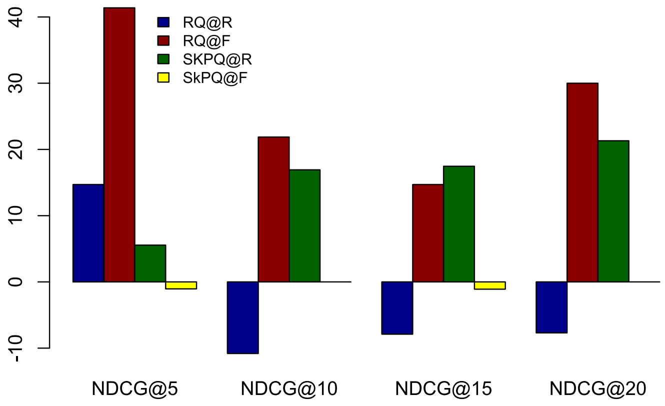


Figure 5.3 Relative NDCG improvements.

Figure 5.3 illustrates the relative NDCG improvement (as described in (SONG et al., 2016)) of the proposed approach e_{pro} over respective baseline model e_{other}, further measured as

$$(e_{pro} - e_{other})/e_{other} \times 100 \tag{5·4}$$

Figure 5.3 reports the relative NDCG improvement values on the vertical axis. The proposed approach demonstrated different degrees of improvement in different scenarios. It improved SKPQ relative NDCG in 20% when using random keywords (SKPQ@R - NDCG@20) and 40% when RQ used frequent keywords (RQ@F - NDCG@5).

Using the users' ratings obtained from Google Maps, we evaluate if our approach improves the query result. Using random keywords, the hotels presented as query results on SKPQ-LD are more popular among the users than the ones presented by the SKPQ. Using frequent keywords, the query result quality on SKPQ-LD is very similar to the one obtained by the SKPQ. Therefore, our approach does not impose a high penalty over the quality of the query result.

5.5 EXPERIMENT 2: EVALUATING FEATURE SELECTION

In Experiment 2, we used the queries in OpinRank to evaluate the feature selection in SKPQ and SKPQ-LD. Since the OpinRank dataset contains only hotel reviews, we restrict our feature dataset to hotels. All hotels used in this experiment are located in Dubai.

Given the query keywords, the SKPQ returns a list of points of interest whose are near to features relevant to the given query keywords. We desire that SKPQ returns objects whose features have a high aspect rating value. This way, the SKPQ would be selecting good features according to users of TripAdvisor. If there is no relevant feature near a point of interest, the SKPQ query result is empty.

The OpinRank dataset offers 5 textual queries for each aspect rating (total of 25 queries). These textual queries were used as query keywords in SKPQ. However, SKPQ did not find any feature whose textual description matched with the query keywords. The description used in SKPQ was too short and could not describe the feature as needed. Notwithstanding, the SKPQ-LD was able to find textual relevant features. From 25 queries, SKPQ-LD was able to find relevant features in 15 (equals to 60% of all executed queries). The features were retrieved with different degrees of textual relevance. Considering $k = 5$ and 25 as the number of executed queries, the MAP score obtained was 0.46.

Between the 15 relevant query results obtained by SKPQ-LD, we could extract the aspect rating value of few features. Many times, the hotel name in OpinRank dataset was not found in DBPedia or OpenStreetMap. Hence, when SKPQ or SKPQ-LD returns a hotel name that does not appear in the OpinRank dataset we can not retrieve its aspect rating value.

We show examples of textual queries that we could extract rating values, and those we could not, to illustrate this scenario. The queries "nice staff" and "good value" are examples of queries that did not return any relevant objects to the user. The objects textual description in SKPQ and SKPQ-LD was not able to describe these aspects of the hotels. However, the queries "great location", "clean place" and "cozy rooms" returned objects when using SKPQ-LD. Figure 5.4 reports the NDCG values of the query results obtained with these query keywords.

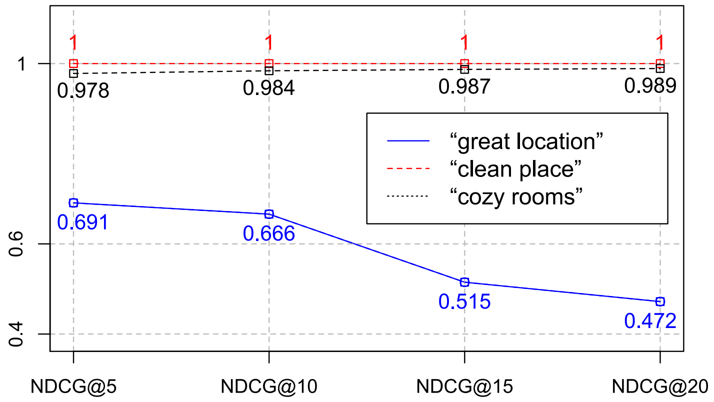


Figure 5.4 SKPQ-LD evaluation using OpinRank.

With the enhancing of objects‘ textual description, SKPQ-LD was able to select more objects that satisfy the user need than SKPQ. Accordingly to the obtained NDCG values in Figure 5.4, SKPQ-LD selected features of good quality. Since the query results have high aspect rating values, we can assume that SKPQ-LD was able to find good objects to the user. For the query “clean place” for example, SKPQ-LD was able to find features that are evaluated by real users as a clean hotel.

The OpinRank dataset contains other queries created by the combination of the queries illustrated in Figure 5.4 plus the queries “nice staff” and “good value”. Nevertheless, the combination of these queries lead to results very similar to the ones at Figure 5.4. In this experiment, the SKPQ-LD demonstrated that the textual description improvement enhances the query capabilities, enabling it to find more objects. Without the textual description improvement, the SKPQ was unable to find any relevant objects to the presented queries.

5.6 DISCUSSION

In this section, we discuss how the textual description enrichment affects the results obtained in ours two experiments and the limitations of our approach. We ran the SKPQ varying query keywords and extend our analysis to understand the difference between textual descriptions from DBpedia and OpenStreetMap. Table 5.4 is an experiment result using hotels from Venice as points of interest (P) and “church” as a query keyword. The first column of Table 5.4 presents the point of interest (POI) textual description, the second column has the point of interest score using traditional SKPQ, and the third column presents the POI’s score using SKPQ-LD. In order to find features that satisfy the spatial selection criteria using LOD, the *geo:geometry* property has to exist in LOD object. For this reason, the point of interest “Palazzo Ferro Fini” and “Splendid Venice” has no score.

Some cities (e.g., Venice) have few spatial objects represented at DBpedia. This database contains only 5 hotels in Venice against 488 registered in OpenStreetMap. Despite the great number of objects, the textual description in OpenStreetMap is poor.

point of interest p	**SKPQ**	**SKPQ-LD**
Hotel Cipriani	0	0.1632
Hotel Danieli	0	0.2789
Grand Hotel des Bains	0	0
Palazzo Ferro Fini	0	no *geo:geometry* property
Splendid Venice	0	no *geo:geometry* property

Table 5.4 Score of object p in Traditional SKPQ Compared With the Score Generated by SKPQ-LD, using hotels from Venice.

While a typical textual description in DBpedia has around 60 terms, the textual description for the same object has only 2 terms in OpenStreetMap. The poor textual description leads the SKPQ to misjudge the evaluation of some points of interest, as can be seen in Table 5.4. Given the query keyword "church", objects "Hotel Cipriani" and "Hotel Danieli" have features in their spatial neighborhood that are textual relevant to the query keyword, but traditional SKPQ fails to identify them because of poor textual description. SKPQ-LD did find these objects and was able to evaluate "Hotel Cipriani" and "Hotel Danieli". For the same reason, SKPQ did not find relevant objects in the Experiment 2 described in Subsection 5.5.

point of interest p	**SKPQ**	**SKPQ-LD**
San Michel Hotel	0.5773	0.25969
Hotel Transamérica	0	0
Hotel Itamarati	0	0.2903
Hotel Braston	0	0.2596
Pousada dos Franceses	0	0.2688

Table 5.5 Score of Object p in Traditional SKPQ Compared With the Score Generated by SKPQ-LD, using hotels from São Paulo.

In order to check whether the problem persists or not, we try hotels in another city. The experiment results using hotels from São Paulo as P objects and "church" as a query keyword was presented in Table 5.5. The column names in Table 5.5 have the same meaning as the column names in Table 5.4. This time we have no problem to find the *geo:geometry* property but SKPQ still has problems to evaluate objects. SKPQ still returns more objects with score zero than SKPQ-LD. These results endorse the improvement obtained by our approach when using random query keywords since "church" is a random query keyword.

As illustrated in Table 5.5, the SKPQ score of the object "San Michel Hotel" is higher than its SKPQ-LD score. When the query keyword has only one term, the textual score takes into account only the length of the document (number of terms) and the term impact. Using traditional SKPQ, we expect a higher object p score than the one computed by SKPQ-LD. The score in traditional SKPQ is higher than SKPQ-LD because the document length is shorter, therefore the term impact in this document is more evident.

5.6.1 Limitations and Points of Improvements

Despite the obtained results look promising, our approach has some limitations. First, although the LOD cloud increases every day, textual descriptions may not always be available with expected quality. This may eventually penalize the query results when using LOD. For instance, the hotel "Splendid Venice" (presented at Section 5.6) does not have the *geo:geometry* property hindering the textual description access by spatial queries.

Zarrinkalam and Kahani (ZARRINKALAM; KAHANI, 2012) describe an enrichment approach using LOD to improve the textual description of articles citations. Accordingly to him, "the Linked Data driven enrichment process has improved the quality of recommendations but it isn't as much as expected" because of "data sources that publish bibliographic information on the LOD cloud, do not yet provide adequately rich and high-quality data, compared to what these data sources provide on the Web of documents".

We face the same problem with spatial information on LOD objects. LinkedGeoData has a higher amount of objects registered than DBpedia. But the textual description of objects in LinkedGeoData is poor as the ones in OpenStreetMap. In addition, a lot of less popular objects are not registered on DBpedia yet or are not well documented. Many objects do not have the *geo:geometry* property too. As a consequence, the textual description of some objects can not be enriched. For this reason, the results obtained by our approach is lower than the ones obtained by the traditional SKPQ when using frequent keywords in Experiment 1. Since the term used as the keyword is frequent in the OpenStreetMap dataset, there is no need for textual description enrichment. If we are looking for objects described as "restaurant" and all restaurants are described in the dataset, there is no need for a more detailed description. The SKPQ performs better in this context because its objects have the description needed and it has access to more objects, so it can search for more restaurants that satisfy the user need.

The world of Linked Data poses many challenges, as described in (GRACIA et al., 2012) and (BIZER et al., 2012). One meaningful challenge is the data integration in the complex and schema-less Semantic Web. However, with the fast growth of the LOD cloud, the semantic annotation becomes more popular and the datasets will provide more quality data. The proposed approach will be even more effective when more high quality data becomes more present in the Web of data.

Chapter

6

This chapter presents the research stages until now. Classes completed and publications are described, as well as, the schedule.

FINAL REMARKS

6.1 ACADEMIC LIFE

In the first two years of the Ph.D. several activities took place in order to define the research topic of the research. The activities include conferences presentations and submission of journal articles. In the early stages the focus was on the Ph.D. classes and curricular components to consolidate the theoretical background. Then, the teaching internship and the workshop presentations to the research group contributed to the communication skill development.

6.1.1 Completed Classes

Each Ph.D. class gave a unique contribution to the research. In the sequence, it is described each of these contributions.

- **MATE64 - Scientific Seminars -** Professor Christina Von Flach organized a series of seminars where several professors or students in the late stage of their research presented interesting topics in computer science. The discussion after these presentations stimulate many ideas and inspired great solutions.
- **MATD74 - Algorithms and Graphs -** This class gave an overview of algorithms complexity and techniques. Professor Tiago de Oliveira Januário presented a series of challenging computational problems which students had to solve using techniques like dynamic programming, recurrences, or greedy algorithms. In addition, a paper entitled "Robot Routing: Genetic Algorithm Applied to Travelling Salesman Problem" was written to demonstrate how to solve an NP-hard problem.
- **MATE66 - Computer Science Research Fundamentals II -** The scientific methodology is one of the most important knowledge to develop solid research. This

class was lectured by Luciano Rebouças de Oliveira who motivated the students to write scientific papers objectively and efficiently. Read and writing methods were studied based on the book “Style - Toward Clarity and Grace” by Joseph M. Williams. As a result, the students produced a paper where the teacher evaluated the students' writing skills. He simulated a journal submission process, thus all papers were evaluated using a blind review method. The paper wrote in this class was the start of the first paper published as a product of this research.

- **MATE32 - Topics on Computer Intelligence II -** Overview of machine learning algorithms focusing on automatic knowledge retrieval from datasets. Professors Ricardo Araújo Rios and Tatiane Nogueira Rios explained how to pre-process data properly and to analyze data using predictive probabilistic methods based on optimization. This class was focused on supervised learning algorithms and how to evaluate them.

- **MATE33 - Topics on Computer Intelligence III -** Professors Ricardo Araújo Rios and Tatiane Nogueira Rios taught about use clustering methods and unsupervised learning algorithms. The students were challenged with clustering problems and led to solving these problems using the main techniques available in the related literature. This class was essential to improve the skill of pattern identification on datasets.

- **MATE85 - Topics on Information Systems and Web I** - This class lectured by the advisor presented core topics on Linked Open Data and Web Semantic. The students developed projects using technologies related to this topic, like Jena and Protégé. This class did a substantial contribution to the research and software development.

- **MATA31 - Oriented Research -** This curricular component is used by the advisor to evaluate the research progress. Periodical feedbacks were given to the advisor who led the research to the correct path with suggestions and contributions.

- **MATA32 - Oriented Teaching Internship -** The advisor offered to the Ph.D. student the experience of teaching to undergraduate students. Additionally, the student gave support for lectures preparation and assisted the class on projects.

- **Classes Dispensed -** Some classes were valuable to the research but they were completed during the master degree on Programa de Pós-Graduação em Computação Aplicada (PGCA). For this reason, the following classes were dispensed by the Programa de Pós-Graduação em Ciência da Computação (PGCOMP):

  - MATE04 - Topics on Databases I
  - MATE10 - Topics on Computer Intelligence I

6.1.2 Publications and Participation in Scientific Conferences

It is important to the Ph.D. student to participate in scientific conferences and to publish the research results on relevant journals. Conferences are a channel to disclose and discuss the research, exchanging information with other researchers from related areas. Following are described the published papers and the conferences attended.

- **WebMedia 2018 - Brazilian Symposium on Multimedia and the Web -** it is the main event of the theme in Brazil and an excellent opportunity for scientific and technical exchange among students, researchers and professionals in the areas of Multimedia, Hypermedia and Web. We published the paper (ALMEIDA; DURÃO, 2018) presenting the preliminary results of our method to enhance the SKPQ accuracy using Linked Open Data.

- **J.UCS 2018 - Journal of Universal Computer Science -** this journal is run by the J.UCS consortium consisting of research institutions from Austria, Germany, Guatemala, USA, and Pakistan. We published the paper (ALMEIDA; DURÃO; COSTA, 2018) detailing our approach to enhance the SKPQ and discussed all obtained results in the experimental evaluation.

- **AMCIS 2019 - Annual Americas Conference on Information Systems -** AMCIS is viewed as one of the leading conferences for presenting the broadest variety of research done by and for information technology academicians. Every year its papers and panel presentations are selected from over 700 submissions, and the AMCIS proceedings are in the permanent collections of libraries throughout the world. From a collaborative work we did a paper about exploiting Web features for relevance feedback. This paper has been accepted and it is in the publishing process.

- **WE.PGCOMP 2018 -** this conference is a curricular component too. After the second year as a Ph.D. student, once per year, the student has to present his research progress to three professors of the doctoral program and to an audience. The research progress is evaluated by the professors who ask questions and make great suggestions for the research.

- **ESWA 2019 - Expert Systems With Applications -** is a refereed international journal whose focus is on exchanging information relating to expert and intelligent systems applied in industry, government, and universities worldwide. The personalization approach we described to improve the SKPQ will be reported in a paper which will be submitted to this journal.

- **Data Mining and Knowledge Discovery 2020 -** the premier technical publication in the field, it is a resource collecting relevant common methods and techniques and a forum for unifying the diverse constituent research communities. We plan to submit a paper to this journal in 2020 describing new techniques and experimental evaluations.

6.2 SCHEDULE

Until now we have proposed two techniques to improve spatial keyword queries accuracy. There were several experimental evaluations demonstrating query improvement. However, there is room to further improve our work. Under those circumstances, Figure 6.1 describes the activities of this research. Activities A01 to A10 was developed before the qualification exam, while A10 to A17 will be developed before the thesis defense.

- **Activity A01 - Academic Classes -** describes the time consumed to complete all classes demanded by the doctoral program. During this time the theoretical background was consolidated. All classes have been described in Section 6.1.1
- **Activity A02 - Literature Review -** studies related to the research topic definition. Many papers were read and drafts were written during this process.
- **Activity A03 - Topic discussion -** meetings with the advisor to discuss research topics and possible solutions to these topics.
- **Activity A04 - Workshops -** meetings involving the research members of the RecSys group. Each meeting consists of members presentation about their research topics followed by practices exercises. There was a workshop about SPARQL queries and another about Jena and Web Semantic.
- **Activity A05 - First proposal: LOD enhancement to spatial queries -** studies related to the research topic definition, writing the proposal definition and define the strategies to solve the problem.
- **Activity A06 - Implementation of the first proposal -** development of the strategies or algorithms defined in the early stage.
- **Activity A07 - Preliminary experiments -** the first rounds of experiments on our approach based on LOD to improve spatial keyword queries.
- **Activity A08 - Evaluation of preliminary results -** analysis of the results with graphs and proper metrics.
- **Activity A09 - Submission to J.UCS 2018 -** the first publication inside the doctoral program. This activity consumed much time because it was the first related to this topic. In this paper, we talked about the benefits of using LOD to enhance textual descriptions of objects and how it improves spatial keyword queries. Several evaluation experiments were described and analyzed.
- **Activity A10 - Submission to WebMedia 2018 -** the first paper submitted to a conference during the doctoral research. Motivated by the late review response on the J.UCS article, in this paper, we described some of the experiments on the LOD enhancement. The conference was a pleasurable experience where was possible to discuss the research topic with experienced researchers.

- **Activity A11 - Extension of the first proposal: query personalization -** after the first publication, the work on the second proposal started. In this activity we worked on the personalization of spatial keyword queries. The experimental evaluation indicates a relevant improvement on these queries accuracy.

- **Activity A12 - Implementation of the proposal extension -** implementation of the strategies or algorithms defined in the early stage.

- **Activity A13 - Preliminary experiments of the proposal extension -** initial experiments on our approach to personalize spatial keyword queries.

- **Activity A14 - Submission to AMCIS 2019 -** in this activity a collaborative work was done, revising the article and attending the demands of the journal's reviewers.

- **Activity A15 - Submission to ESWA 2019 -** the second paper describing the personalization approach is submitted and we are waiting for the journal response.

- **Activity A16 - Qualification -** write qualification text and prepare the qualification presentation.

- **Activity A17 - New optimization research -** explore new methods to improve spatial keyword queries. New algorithms or hybrid solutions can be proposed in this stage.

- **Activity A18 - Implementation of the new optimization -** this activity will implement the strategies or algorithms defined in the previous activity.

- **Activity A19 - Experiment results -** after the development stage it is necessary to evaluate the algorithms or method developed.

- **Activity A20 - Experiment analysis -** the experimental results are analyzed and compared with previous results, as well as, with the results reported in the literature.

- **Activity A21 - Preliminary results paper -** another paper will be published introducing the new approach and the results obtained so far. It is possible to submit on paper to a journal and another to a conference in this stage.

- **Activity A22 - Thesis -** activity defined to write the thesis.

- **Activity A23 - Thesis defense -** designated to thesis presentation and research end.

The schedule of activities executed since the course enrollment together with the ones that will be executed after the qualification exam are presented in Figure 6.1.

Date		Activities																						
Year	Month	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23
2016	Nov	■	■		■																			
	Dec	■	■		■																			
2017	Jan	■	■	■																				
	Feb	■	■	■																				
	Mar	■	■	■																				
	Apr	■	■	■		■																		
	May	■				■	■																	
	June	■					■	■		■														
	July	■					■	■		■														
	Aug	■					■	■		■														
	Sept	■					■		■	■														
	Oct	■							■	■														
	Nov	■							■	■														
	Dec	■							■	■														
2018	Jan	■								■														
	Feb	■								■														
	Mar											■												
	Apr											■	■											
	May												■											
	June												■											
	July										■		■	■	■									
	Aug										■		■	■	■									
	Sept										■		■	■	■									
	Oct										■		■	■	■									
	Nov												■	■	■									
	Dec												■	■	■	■								
2019	Jan												■	■	■	■								
	Feb													■	■	■								
	Mar													■	■	■	■							
	Apr													■	■	■	■							
	May															■	■							
	June															■	■							
	July															■	■							
	Aug																	■						
	Sept																	■	■					
	Oct																	■	■					
	Nov																	■	■					
	Dec																	■	■					
2020	Jan																	■	■			■	■	
	Feb																	■	■	■		■	■	
	Mar																	■	■	■	■	■	■	
	Apr																				■	■	■	
	May																						■	
	June																						■	
	July																						■	
	Aug																						■	
	Sept																							■
	Oct																							■

Figure 6.1 Ph.D. activities since the course enrollment until thesis defense.

6.3 SUMMARY

The aforementioned chapter described the student life academy until now. It was depicted the classes completed and how they contributed to the research. In addition, it is possible to see the timeline between the publications and the stages before them. On the other hand, the schedule illustrates the time left to the thesis defense and future works followed by each scheduled date.

BIBLIOGRAPHY

ADAMS, B. From spatial representation to processes, relational networks, and thematic roles in geographic information retrieval. In: ACM. *Proceedings of the 12th Workshop on Geographic Information Retrieval.* [S.l.], 2018. p. 1.

ADELEKE, A. O. et al. A group-based feature selection approach to improve classification of holy quran verses. In: SPRINGER. *International Conference on Soft Computing and Data Mining.* [S.l.], 2018. p. 282–297.

ALLAN, J. et al. Frontiers, challenges, and opportunities for information retrieval: Report from swirl 2012 the second strategic workshop on information retrieval in lorne. In: ACM. *ACM SIGIR Forum.* [S.l.], 2012. v. 46, n. 1, p. 2–32.

ALMEIDA, J. P. D. de; DURÃO, F. A. Improving the spatial keyword preference query with linked open data. In: SBC. *Anais Estendidos do XXIV Simpósio Brasileiro de Sistemas Multimídia e Web.* [S.l.], 2018. p. 19–24.

ALMEIDA, J. P. D. de; DURÃO, F. A.; COSTA, A. F. da. Enhancing spatial keyword preference query with linked open data. v. 24, n. 11, p. 1561–1581, nov 2018.

ALMEIDA, J. P. D. de; ROCHA-JUNIOR, J. B. Top-k spatial keyword preference query. *JIDM*, v. 6, n. 3, p. 162–177, 2015.

ALMEIDA, J. P. D. de; ROCHA-JUNIOR, J. B. Top-k spatial keyword preference query. *Journal of Information and Data Management*, p. 162, 2016.

BALTRUNAS, L.; MAKCINSKAS, T.; RICCI, F. Group recommendations with rank aggregation and collaborative filtering. In: ACM. *Proceedings of the fourth ACM conference on Recommender systems.* [S.l.], 2010. p. 119–126.

BARUFFOLO, A. R-trees for astronomical data indexing. In: *Astronomical Data Analysis Software and Systems VIII.* [S.l.: s.n.], 1999. v. 172, p. 375.

BATTLE, R.; KOLAS, D. Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, p. 355–370, 2011.

BAYER, R.; MCCREIGHT, E. Organization and maintenance of large ordered indexes. In: ACM-SIGFIDET. *Workshop on Data Description and Access.* Houston, Texas, 1970.

BECKER, C.; BIZER, C. Exploring the geospatial semantic web with dbpedia mobile. *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, p. 278–286, 2009.

BECKMANN, N. et al. The R*-tree: An efficient and robust access method for points and rectangles. In: *SIGMOD*. [S.l.]: ACM, 1990. p. 322–331.

BELESIOTIS, A. et al. Spatio-textual user matching and clustering based on set similarity joins. *The VLDB Journal—The International Journal on Very Large Data Bases*, Springer-Verlag, v. 27, n. 3, p. 297–320, 2018.

BENTLEY, J. L.; FRIEDMAN, J. H. Data structures for range searching. *ACM Computing Surveys (CSUR)*, ACM, v. 11, n. 4, p. 397–409, 1979.

BERNERS-LEE, T. Design issues: Linked data (2006). *URL http://www.w3.org/DesignIssues/LinkedData.html*, 2011.

BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. Rfc 3986. *Uniform Resource Identifier (URI): Generic Syntax*, InternetEngineering Task Force, 2005.

BERNERS-LEE, T. et al. The semantic web. *Scientific american*, New York, NY, USA:, v. 284, n. 5, p. 28–37, 2001.

BIZER, C. et al. The meaningful use of big data: four perspectives–four challenges. *ACM SIGMOD Record*, ACM, v. 40, n. 4, p. 56–60, 2012.

BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, p. 205–227, 2009.

BORZSONY, S.; KOSSMANN, D.; STOCKER, K. The skyline operator. In: IEEE. *Data Engineering, 2001. Proceedings. 17th International Conference on.* [S.l.], 2001. p. 421–430.

BOUHANA, A. et al. An ontology-based cbr approach for personalized itinerary search systems for sustainable urban freight transport. *Expert Systems with Applications*, Elsevier, v. 42, n. 7, p. 3724–3741, 2015.

BRAUN, M.; SCHERP, A.; STAAB, S. Collaborative creation of semantic points of interest as linked data on the mobile phone. In: *Extended Semantic Web Conference (Demo Session)*. [S.l.]: Springer, 2010.

BÜTTCHER, S.; CLARKE, C. L.; CORMACK, G. V. *Information retrieval: Implementing and evaluating search engines.* [S.l.]: Mit Press, 2016.

CAO, X. et al. Spatial keyword querying. In: *ER*. [S.l.]: Springer, 2012. p. 16–29.

CARMEL, D.; GUETA, G.; BORTNIKOV, E. *Top-k query processing with conditional skips.* [S.l.]: Google Patents, 2018. US Patent App. 15/345,277.

CHEN, L. et al. Spatial keyword query processing: an experimental evaluation. In: . [S.l.]: VLDB Endowment, 2013. p. 217–228.

CHEN, Z. et al. Distributed publish/subscribe query processing on the spatio-textual data stream. In: IEEE. *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. [S.l.], 2017. p. 1095–1106.

CHOMICKI, J. Preference formulas in relational queries. *ACM Transactions on Database Systems (TODS)*, ACM, v. 28, n. 4, p. 427–466, 2003.

COHEN, W. W.; RAVIKUMAR, P.; FIENBERG, S. E. A comparison of string distance metrics for name-matching tasks. *Workshop on Data Cleaning and Object Consolidation*, p. 73–78, 2003.

COMER, D. Ubiquitous b-tree. *ACM Computing Surveys (CSUR)*, ACM, v. 11, n. 2, p. 121–137, 1979.

CONG, G.; JENSEN, C. S.; WU, D. Efficient retrieval of the top-k most relevant spatial web objects. In: . [S.l.]: VLDB Endowment, 2009. v. 2, n. 1, p. 337–348.

DAI, J. et al. On personalized and sequenced route planning. *World Wide Web*, Springer, v. 19, n. 4, p. 679–705, 2016.

DANGERMOND, J. *Spatial Thinking Is Fundamental.* 2017. Disponível em: <https://www.forbes.com/sites/esri/2017/11/02/spatial-thinking-is-fundamental/\#18c24c2e7aab>.

DIETZ, J. L. *What is Enterprise Ontology?* [S.l.]: Springer, 2006.

FERNÁNDEZ-TOBÍAS, I. et al. A generic semantic-based framework for cross-domain recommendation. In: ACM. *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems.* [S.l.], 2011. p. 25–32.

FIELDING, R. et al. *Hypertext transfer protocol–HTTP/1.1.* [S.l.], 1999.

FINKELSTEIN, L. et al. Placing search in context: The concept revisited. *ACM Transactions on information systems*, v. 20, n. 1, p. 116–131, 2002.

FRESSATO, E. P. *Incorporação de metadados semânticos para recomendação no cenário de partida fria.* 105 p. Tese (Doutorado) — Universidade de São Paulo, 2019.

GANESAN, K.; ZHAI, C. Opinion-based entity ranking. *Information Retrieval*, 2011.

GANTZ, J.; REINSEL, D. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, v. 2007, n. 2012, p. 1–16, 2012.

GASPARETTI, F. Personalization and context-awareness in social local search: State-of-the-art and future research challenges. *Pervasive and Mobile Computing*, Elsevier, v. 38, p. 446–473, 2017.

GRACIA, J. et al. Challenges for the multilingual web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, v. 11, p. 63–71, 2012.

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, Elsevier, v. 43, n. 5-6, p. 907–928, 1995.

GUO, S.; ALAMUDUN, F.; HAMMOND, T. Résumatcher: A personalized résumé-job matching system. *Expert Systems with Applications*, Elsevier, v. 60, p. 169–182, 2016.

GÜTING, R. H. An introduction to spatial database systems. *The VLDB Journal–The International Journal on Very Large Data Bases*, Springer-Verlag New York, Inc., v. 3, n. 4, 1994.

GUTTMAN, A. *R-trees: a dynamic index structure for spatial searching.* [S.l.]: ACM, 1984. v. 14.

GUTTMAN, A.; STONEBRAKER, M. Using a relational database management system for computer aided design data. *IEEE Database Eng. Bull.*, v. 5, n. 2, p. 21–28, 1982.

HAGEN, P.; MANNING, H.; SOUZA, R. Smart personalization. *Forrester Research, Cambridge, MA*, 1999.

HARIHARAN, R. et al. Processing spatial-keyword (sk) queries in geographic information retrieval (gir) systems. In: IEEE. *Scientific and Statistical Database Management, 2007. SSBDM'07. 19th International Conference on.* [S.l.], 2007. p. 16–16.

HEARST, M. A. 'natural'search user interfaces. *Communications of the ACM*, ACM, v. 54, n. 11, p. 60–67, 2011.

HEGDE, V. et al. Utililising linked data for personalized recommendation of poi's. In: *International AR Standards Meeting, Barcelona, Spain.* [S.l.: s.n.], 2011.

HUTT, K. A comparison of rdf query languages. In: *Proc. of 21th Computer Science Seminar, Hartfort, Connecticut.* [S.l.: s.n.], 2005. p. 1–7.

ILYAS, I. F.; BESKALES, G.; SOLIMAN, M. A. A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys (CSUR)*, ACM, v. 40, n. 4, p. 11, 2008.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 20, n. 4, p. 422–446, 2002.

KARAM, R.; MELCHIORI, M. Improving geo-spatial linked data with the wisdom of the crowds. In: ACM. *Proceedings of the joint EDBT/ICDT 2013 workshops.* [S.l.], 2013. p. 68–74.

KARPATHIOTAKI, M. et al. Prod-trees: semantic search for earth observation products. In: SPRINGER. *European Semantic Web Conference.* [S.l.], 2014. p. 374–378.

KELES, I. *Spatial Keyword Querying: Ranking Evaluation and Efficient Query Processing*. Tese (Doutorado) — Aalborg Universitetsforlag, 2018.

KHODAEI, A.; SHAHABI, C.; LI, C. Hybrid indexing and seamless ranking of spatial and textual features of web documents. In: SPRINGER. *Database and Expert Systems Applications*. [S.l.], 2010. p. 450–466.

KIESSLING, W. Foundations of preferences in database systems. In: VLDB ENDOWMENT. *Proceedings of the 28th international conference on Very Large Data Bases*. [S.l.], 2002. p. 311–322.

KLYNE, G.; CARROLL, J. J. Resource description framework (rdf): Concepts and abstract syntax. 2006.

KWON, O.; SHIN, M. K. Laco: A location-aware cooperative query system for securely personalized services. *Expert Systems with Applications*, Elsevier, v. 34, n. 4, p. 2966–2975, 2008.

LACROIX, M.; LAVENCY, P. Preferences; putting more knowledge into queries. In: *VLDB*. [S.l.: s.n.], 1987. v. 87, p. 1–4.

LAURINI, R.; THOMPSON, D. *Fundamentals of spatial information systems*. [S.l.]: Academic press, 1992.

LI, G. et al. Reverse top-k query on uncertain preference. In: SPRINGER. *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. [S.l.], 2018. p. 350–358.

LI, Z. et al. Ir-tree: An efficient index for geographic document search. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 23, n. 4, p. 585–599, 2011.

LIU, J. et al. Clue-based spatio-textual query. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 10, n. 5, p. 529–540, 2017.

LUCCHESE, C. et al. Exploiting cpu simd extensions to speed-up document scoring with tree ensembles. In: ACM. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. [S.l.], 2016. p. 833–836.

MACKENZIE, J.; CHOUDHURY, F. M.; CULPEPPER, J. S. Efficient location-aware web search. In: ACM. *Proceedings of the 20th Australasian Document Computing Symposium*. [S.l.], 2015. p. 4.

MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. An introduction to information retrieval. *Natural Language Engineering*, Cambridge university press, v. 16, n. 1, p. 100–103, 2010.

MANNING, C. D. et al. *Introduction to information retrieval*. [S.l.]: Cambridge university press, 2008. v. 1.

MARGARIS, D.; VASSILAKIS, C.; GEORGIADIS, P. Query personalization using social network information and collaborative filtering techniques. *Future Generation Computer Systems*, Elsevier, v. 78, p. 440–450, 2018.

MCCRAE, J. P. *The Linked Open Data Cloud.* 2019. Disponível em: <https://lod-cloud.net/>.

MENG, X.; ZHANG, X.; ZHAO, Z. T k qs: A top-k keyword query suggestion system. In: IEEE. *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).* [S.l.], 2018. p. 1005–1008.

NIKOLAOU, C. et al. Sextant: browsing and mapping the ocean of linked geospatial data. In: SPRINGER. *Extended Semantic Web Conference.* [S.l.], 2013. p. 209–213.

PAN, J. Z. Resource description framework. In: *Handbook on ontologies.* [S.l.]: Springer, 2009. p. 71–90.

PAPADIAS, D. et al. Efficient OLAP operations in spatial data warehouses. In: *SSTD.* [S.l.]: Springer, 2001. p. 443–459.

PÉREZ, J.; ARENAS, M.; GUTIERREZ, C. Semantics and complexity of sparql. *ACM Transactions on Database Systems (TODS)*, ACM, v. 34, n. 3, p. 16, 2009.

PERRY, M.; HERRING, J. Ogc geosparql-a geographic query language for rdf data. *OGC Implementation Standard. Sept*, 2012.

PURVES, R. S. et al. Geographic information retrieval: Progress and challenges in spatial search of text. *Foundations and Trends® in Information Retrieval*, Now Publishers, Inc., v. 12, n. 2-3, p. 164–318, 2018.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. *Spatial databases: with application to GIS.* [S.l.]: Morgan Kaufmann, 2001.

ROCHA-JUNIOR, J. B. *Efficient processing of preference queries in distributed and spatial databases.* Tese (Doutorado) — Norwegian University of Science and Technology, 2012.

ROCHA-JUNIOR, J. B. et al. Efficient processing of top-k spatial keyword queries. In: *SSTD.* [S.l.]: Springer, 2011. p. 205–222.

SALMINEN, A.; TOMPA, F. W. Pat expressions: an algebra for text search. *Acta Linguistica Hungarica*, v. 41, n. 1, p. 277–306, 1994.

SAMET, H. The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, ACM, v. 16, n. 2, p. 187–260, 1984.

SAQUICELA, V. et al. Lod-gf: An integral linked open data generation framework. In: SPRINGER. *Conference on Information Technologies and Communication of Ecuador.* [S.l.], 2018. p. 283–300.

SARKER, M. K. et al. Explaining trained neural networks with semantic web technologies: First steps. *arXiv preprint arXiv:1710.04324*, 2017.

SEO, Y.-D. et al. An enhanced aggregation method considering deviations for a group recommendation. *Expert Systems with Applications*, Elsevier, v. 93, p. 299–312, 2018.

SHANBHAG, A.; PIRK, H.; MADDEN, S. Efficient top-k query processing on massively parallel hardware. In: ACM. *Proceedings of the 2018 International Conference on Management of Data.* [S.l.], 2018. p. 1557–1570.

SKORKOVSKÁ, L. Relevant documents selection for blind relevance feedback in speech information retrieval. In: SPRINGER. *International Conference on Text, Speech, and Dialogue.* [S.l.], 2016. p. 418–425.

SONG, H. et al. Individual judgments versus consensus: Estimating query-url relevance. *ACM Transactions on the Web (TWEB)*, ACM, v. 10, n. 1, p. 3, 2016.

SUGIURA, A.; ETZIONI, O. Query routing for web search engines: Architecture and experiments. *Computer Networks*, Elsevier, v. 33, n. 1-6, p. 417–429, 2000.

THOMPSON, P. et al. Enriching news events with meta-knowledge information. *Language Resources and Evaluation*, Springer, v. 51, n. 2, p. 409–438, 2017.

TRIPERINA, E. et al. Creating the context for exploiting linked open data in multidimensional academic ranking. *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, v. 3, n. 3, p. 33–43, 2015.

VARUDHARAJULU, A. K.; MA, Y. Feature-based facebook reviews process model for e-management using data mining. In: ACM. *Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning.* [S.l.], 2019. p. 406–410.

WANG, J. et al. Diversionary comments under blog posts. *ACM Transactions on the Web (TWEB)*, ACM, v. 9, n. 4, p. 18, 2015.

WU, D.; CONG, G.; JENSEN, C. S. A framework for efficient spatial web object retrieval. *The VLDB Journal–The International Journal on Very Large Data Bases*, Springer-Verlag New York, Inc., v. 21, n. 6, p. 797–822, 2012.

WU, D. et al. Joint top-k spatial keyword query processing. *Knowledge and Data Engineering, IEEE Transactions on*, IEEE, v. 24, n. 10, p. 1889–1903, 2012.

XU, J.; CROFT, W. B. Quary expansion using local and global document analysis. In: ACM. *Acm sigir forum.* [S.l.], 2017. v. 51, n. 2, p. 168–175.

YANG, Z.; MOFFAT, A.; TURPIN, A. How precise does document scoring need to be? In: SPRINGER. *Asia Information Retrieval Symposium.* [S.l.], 2016. p. 279–291.

YIU, M. L. et al. Top-k spatial preference queries. In: IEEE. *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on.* [S.l.], 2007. p. 1076–1085.

YU, L. Linked open data. In: *A Developer's Guide to the Semantic Web.* [S.l.]: Springer, 2011. p. 409–466.

ZARRINKALAM, F.; KAHANI, M. A multi-criteria hybrid citation recommendation system based on linked data. In: IEEE. *Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on.* [S.l.], 2012. p. 283–288.

ZHOU, Y. et al. Hybrid index structures for location-based web search. In: ACM. *Proceedings of the 14th ACM international conference on Information and knowledge management.* [S.l.], 2005. p. 155–162.

ZHU, S. et al. Scaling up top-K cosine similarity search. *Data & Knowledge Engineering*, Elsevier, p. 60–83, 2011.

ZOBEL, J.; MOFFAT, A. Inverted files for text search engines. In: . [S.l.]: ACM, 2006. p. 6.

Appendix

A

USER PROFILE BASED ON ROOM ASPECT VALUE

@relation room

@attribute class {1,5}

@attribute description string

@data

5,'Best location and accommodation of the Jumeirah Properties We have just returned from a week stay at the beit al bahar villas. The accomodation, service, and faclities were excellent. We did however get upgraded from a 1bed villa to a 2bed villa due to construction in front of our patio. The difference was huge. The 2 bedroom villa has a far larger patio with 4 deckchairs and over looked the beach and the burj. The 1 bed villa was located on the first row with a view onto the nearby villas- pretty disappointing. So make sure if you stay here you request a villa in the 3rd row- those with views over the beach and the burj al arab.I have stayed at the mina and jumeirah beach hotel and these villas are the best we have stayed in. They have an arabic feel and with an extensive patio with loungers and a plunge pool- you couldnt get better accomodation.Each villa has a huge living room - L-shaped sofa and a separate table with 4chairs. You also get a latop with free internet access. The bathroom is amazing with the largest and deepest bath I have ever seen- takes around 30mins to fill up!You also have access to the private beach for execuvite/premium/madinat/burj guests. Also, there is a pool just for villa guests with a small outdoor bar/restaurant in the middle of the villa area. There is also a villa restaurant just for villa guests which we only went to for breakfast.The best thing is that you dont need to walk through a large hotel to get to the beach/pool- u just leave ur villa and u are there. It makes it a much more relaxing holiday.'

5,'Our favourite place in Dubai We had a fabulour time here. The privacy is the best, relaxing in your garden with a plunge pool, brought sandwiches at 4.00, drinks at the beautiful bar/pool at 7.00. The two bedroom villa we had was very spacious and I love the Arabian decor, staff all very attentive and call you by name. A little name card is put up on the wall outside, all nice touches to make you feel welcome and at home. It is great being able to call the buggies especially for BAB guests. Bulgari toiletries in both bathrooms, internet area, lounge and long hall. The Burj is fab too but the being able to sit outside just gives it the edge. Cant wait to return, if you go here you will not be disappointed.'

5,'Wow This Is Dubai At Its Best! Four of us stayed for a week in March 2007 at the Beit Al Bahar Villas, what an experience simply incredible beautiful villas sumptiously furnished and waited on hand and foot, This is the place to be pampered. The only interruptions we had were the helicopters landing on the helipad at the Burj . If you want a wonderfully relaxing stay in Dubai book these villas, nothing to compare. Margaret Swansea Wales.'

5,'Absolute perfection! The property is absolutely beautiful; we loved all the traditional Arabic accents and decorations as well as the comforting environment in the room. We were able to fully &amp; completely relax while at our villa and at the beach. We were so comfortable and relaxed we didn’t even worry about locking our doors while we were out or locking up our cameras while we swam in the ocean. Not having to worry about the little things, made a big difference to our trip. There are so many examples of the Beit Al Bahar and the Jumeriah Beach Hotel staff ensuring we didn’t have to worry or think about anything. The only time we really had to think, was when we tried to decide where to eat at the many fantastic Jumeriah International restaurants! The Beit Al Bahar staff was always so kind and helpful and they truly anticipated our needs, sometimes before we even knew what our needs were! One afternoon after a day on the beach, we were in our villa debating going to get a light snack as we didn’t have dinner reservations until 10pm. Literally, during this discussion, our doorbell rang and the villa staff brought us afternoon tea. This was completely unexpected and was appreciated for many reasons – anticipating our needs, a light snack and most importantly – a traditional experience. When we sadly left Dubai, the item which stood out from our trip the most was the people. Every single person we interacted with was extremely friendly, kind and helpful. We have never been surrounded by such hospitality and such wonderful people! We have stayed at other “The Leading Hotels of the World”, while they all live up to a higher standard; the Jumeriah Beach Hotel exceeds this standard.'

5,'The Ultimate The Beit Al Bahar villas are simply stunning. We have stayed here on 2 occassions, and both times have been completey overwhelmed by the sheer luxury and levels of service. My husband had a days sea fishing from the hotel - on his return a member of the villas staff was waiting in a golf buggy at the marina with cold face towels and freezing water ready to take him back to our villa. Our 7 year old daughter had a great time - again the staff whisked her off in a buggy to the kids club and picked her up when she was ready, also the wild and wadi water park is on your doorstep and you get priority entry at all times. I really can not think of one negative thing to say about this place - it is pure luxury!!Although terribly expensive, I would always hope we will be fortunate enought to go back again.'

5,'What a wonderful hotel - top luxury Had a fantastic stay here for 7 days last Sept.Lovely hotel, very relaxed place great to just chill out. Fantastic pool and leisure facilities and very spacious rooms with great shower / wet room.Great restaurants and top quality food.Service in all areas was top and all the staff went that extra mile to try and assist you.My only complaint about Dubai was that the availability of taxis was a nightmare, maybe because it was at the end of Ramandan, we often had to wait for 45 minutes plus outside some shopping malls to get a taxi. It needs to get this sorted asap if it is going to compete with other major cities.This hotel is top notch also with quality furniture and fittings throughout.The Burj al Arab is all glitz and no substance in comparison.I would definately pay that bit extra and stay at the Residence &amp; Spa.LightingMan

5,'Simply the best My family and i stayed at the mirage residence after staying at almost all the jumeira beach hotels.I must say we are always going to stay here from now on. Anyone who has been to dubai will tell you that even the 3 star hotel standards are comparable to 4-5 star hotel in other countries.The residence is definately for the traveler that does not want to be bothered by noise or families with no regard or respect for the peace of other travelers. I was glad that only residence guests had the exclusive use of the dedicated pool area, But also loved the fact that we could use all the facilities of the Arabian court and palace. Staff are the best and usually address you by your name which is a great touch.perfect hotel and well worth the price you pay

5,'Beautiful Hotel I highly recommend this hotel. We got waited on hand and foot and could not have asked for a better hotel stay. This hotel and all the staff made our holiday magical.It is true what they say...you get what you pay for!!'

5,'Again another magical stay ..... This is the best hotel in Dubai, that s why we return, year after year!!The simple personal touches, attention to detail and shere luxury, the Residence

&amp; Spa has it all, 400%!We travelled with my parents and had an interconnecting Prestige Room &amp; Junior suite. Great for travelling with kids as they have space to cool down during noon, as we went inside then because of the heat.Breakfast at the Dining Room - delicious, dinner at the Dining Room - superb, dinner at the Rotisserie (Arabian Court) - plenty of choice for the kiddies and delicious, dinner at Nina - the Indian place to dine!Menu at both the Dining Room and The Rotisserie should be changed a little more often as the same dishes turned up. But then again, not many guests stay for 12 nights .....Afternoon tea at the Library: scumptious!!!And yes, use the well equipped gym at the Spa after that :-))))))Oh, you like to go out? Check out the Kasbar (next to the Palace)!!!! Expensive drinks but great nightclub, even for us (in our forties with 3 kids - hahahaha!).Many thanks again to Mr Philippe, Fikri, Roshan, Mary (KidsOnly), Raj (pool), The Dining room Chef, and all other staff who made our 12 night stay memorable again.We shall return, soon ....'

5,'absolutely fantastic   The residence and spa is a hotel to spend a week indulging in absolute luxury and quiet.The staff are brilliant and make you feel special, no request is too much for them.'

1,'Good and bad   Overall our experience was OK. The two bedroom apartment we had was pretty clean, but everything needed updated. Leaks in the plumbing, washing machine did not work. Electrical problems. TV remotes were missing the backs. The windows were filthy. Very difficult to get a cab. When the hotel called a cab for you, you were over charged quite a bit. I was charged for an extra day, even though I had called three days prior to cancel the first day. Most of the staff were nice and helpful. Location of the hotel is in a bad part of Dubai, but the price was very good compared to other more modern parts of Dubai. Music late at night from the hotel bar was loud and could be heard in the rooms. Personally, I would not stay there again, but if you dont mind an older hotel with a lousy restaurant, I would say give it a try.'

1,'To be avoided at all costs   We stayed here last week, last stop after Australia. It was overall a very negative experience, our lst room was surrounded by building sites, noise overwhelming so we asked to be changed. Our next room was next to the mosque so we were awoken at an unearthly hour! The rooms were spacious but not clean, the hood of the cooker badly burnt, only one cup! no complimentary tea or coffee or course. All the rooms had a musty smell. The staff were variable, certainly not much of a welcome. The complimentary airport pickup had to be paid for. The bedding smelt of smoke, probably our worst hotel stay. This was more expensive than Ramee Hotel Aptments, which werent much good either but a slight improvement. This is a very mediochre chain, and just not worth it, much better to pay a bit more.'

1,'A better hotel would have been worth every Dirham!   I counted a total of two staff people on arrival at 8pm. The receptionist and the Bellman. The AC wasnt working in the first room we were assigned to... it took too long for a maintenance person to show-up, so we packed our bags and were ready to leave before they moved our room. After we requested toilet paper, the Bellman came with an un boxed, unwrapped bunch of facial tissue. Imagine the joy! Apparently, housekeeping was gone for the night and noone could get into the supplies. The following evening, the wait for toilet paper turned from a promise of five minutes to a 45 minute wait. The beds are worn beyond use... but theyre still there. The bathrooms had a stench coming from the drainage pipes in the floor. After the first night, went upstairs to the top floor to check out the breakfast.... youll find that the breakfast room is a room off to the side of the GYM! You have to go into the gym to have breakfast. The whole deal didnt seem sanitary, so we passed on the breakfast offerings during our stay. The Ramee Group is a pretty large, &quot;middle market&quot; chain, and Ive stayed at an Ramee in Bahrain (minus the negative experience), but clearly, the qulaity of the property and service is not consistent and is lacking heavily at the Guestline Apartments II in Dubai.'

1,'Okay if you need the space We were in Dubai for 3 nights and have 2 children, both under the age of two. The Ramee II was good in that it has a seperate master bedroom and a large living/kitchen area. So with children, it was good to have the space.I have to agree with other reviews, however, that the Ramee II is not worth returning to. The rooms were not very clean and we too had the construction site right outside our window. Sometimes they worked until 11PM! It was only the fact that the jackhammers drowned out the sound of our screaming children that I didnt mind. The reception staff was friendly, but overall not helpful. We paid an extra $25/night for each child which was a disgrace since they didnt even provide extra linen for the children. (Im debating this with Expedia at the moment.) Plus, we were told that it would be no problem to have 2 port-a-cots (cribs) for the children and when we arrived there was one, very unsafe, wooden bassinet. Either of my children would have easily injured themselves had we used it. Without an alternative, the children slept with us. So it was no holiday.We booked this hotel at a sale price through Expedia. I would NEVER pay full price for a room in this hotel and would consider it to be about 3-3.5 star, not 4.'

1,'Horrible Terrible and Horrible again Having had the great misfortune to book this hotel for 2 weeks in January 2007 i feel it is my duty to warn anyone else to try every other hotel before booking here. I could normaly look past the large but filthy rooms, some of the unfriendliest reception staff i have ever experienced and the fact that the place was not cleaned in all my time there.. But the fact that they never warned me that we would be woken up at 6:30 EVERY MORNING by the sound of hammers, drills and cranes from the building site next door was just sneaky..Even when i politely asked to be transfered to one of many spare rooms they had, facing away from the noise at least ,my request was not so politely refused. To sum up: pay $120 + for this place only if everywhere else in Dubai was closed...otherwise try one of the other Cleaner hotels around Bur Dubai

1,'Dark rooms Stayed here for 2 weeks (booked initially for 4). The location is still pretty much a building site across dubai internet city, next to street 611 in The Greens. The views are either onto the next block, the building site or Sheik Zayed Road - so not really appealing.We booked a 1 room apartment - the pictures of the hotel on their web site are for real: the rooms are soo dark, you have to have the lights on during the whole day. The airco was so noisy and windy, that we had to turn it off to prevent headaches.The room service was ok, they even cleaned the dirty dishes.If you have to stay on a budget (around 16,000AED per month) and dont mind looking on a building site/busy street, it might be ok

1,'Avoid it if you can This is a very disappointing hotel and represents poor value for money. Okay, it might be less expensive by normal Dubai standards but you cant help getting that ripped off feeling. I have read other reviews that said the rooms are small. That is an understatement. It appears clean but is badly in need of basic maintenance and decoration. The basin in our bathroom was blocked and we had to share with a cockroach but the hot water was plentiful. Most staff were friendly and helpful but one was quite the opposite. Our advice would be pay a little extra and avoid this one. It has the potential to take the shine off your holiday.'

1,'Don t go here This hotel was very disappointing. Despite advertising airport pick-up our request was totally ignored. We found it impossible to stay in the first room we were allocated. It was very stuffy and the air conditioning which was nearly non-existent did not do anything to help. The window had been screwed down and was impossible to open. We asked to change rooms and the second room was inhabitable despite the cracked sink and leaking bath!. A fellow guest with the same problem decided not to stay at all and checked out immediately to go and find another hotel. The working staff were anxious to help us, especially in the restaurant where the food was good value for money, but the management was very poor. This was an experience we would not wish to repeat.'

1,'worlds worst pick-up joint.   This is the worst place ever. It is a dirty disgusting pick-up joint!As soon as you walk into the bar after about 6pm, you will be visited by numerous amounts of woman, asking you if you have any requests!!!!!! If you know what i mean.The rooms were disgusting, the music from the club below bellowed until the early hours, filling the whole hotel with unbearable noise. If you are single, have little or no money, and dont mind sleeping it (totally) rough in Dubai, then this is the place for you!'

1,'(--) up place One of the lousiest hotel we have stayed in our life. The front desk lady (Ms. Sunita) was kind enough to give us two room s- one 1 hr. late, another 3.5 hr late after stipulated time. No water to drink, you have to buy water bottle and pay 6 dirhams instead of 1 dirhams. Breakfast is pathetic, stale bread, instead of Juice some Tang, fruits only watermelon, old vegetables. The dirty hotel with full of mouse. The room boys are unhelpful and only try to fleece you. The bar is also bad place with only fat aunties gyrating to some hindi songs.. So if you want to experience hell , liked to be cheated and enjoy the fleecing by others please check into this hotel and ask for Ms. Sunita.'