



Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-Graduação em Ciência da Computação

**A FRAMEWORK FOR EXPLOITING OPEN
DATA TO IMPROVE SPATIAL KEYWORD
QUERY APPLICATIONS**

João Paulo Dias de Almeida

TESE DE DOUTORADO

Salvador
3 de maio de 2021

JOÃO PAULO DIAS DE ALMEIDA

**A FRAMEWORK FOR EXPLOITING OPEN DATA TO IMPROVE
SPATIAL KEYWORD QUERY APPLICATIONS**

Esta Tese 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
3 de maio de 2021

Sistema de Bibliotecas - UFBA

Almeida, João Paulo Dias de.

A Framework for Exploiting Open Data to Improve Spatial Keyword Query Applications / João Paulo Dias de Almeida – Salvador, 2021.
167p.: il.

Orientador: Prof. Dr. Frederico Araújo Durão.

Tese (Doutorado) – Universidade Federal da Bahia, Instituto de Matemática, 2021.

1. Spatial Data. 2. Spatial Query. 3. Linked Open Data. 4. Query Evaluation. 5. Query Processing. 6. Personalization. I. Durão, Frederico Araujo. II. Universidade Federal da Bahia. Instituto de Matemática. III Título.

CDD – XXX.XX

CDU – XXX.XX.XXX

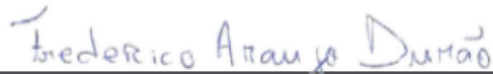
TERMO DE APROVAÇÃO

JOÃO PAULO DIAS DE ALMEIDA


A FRAMEWORK FOR EXPLOITING OPEN DATA TO IMPROVE SPATIAL KEYWORD QUERY APPLICATIONS

Esta Tese de Doutorado foi julgada adequada à obtenção do título de Doutor em Ciência da Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia.

Salvador, 03 de maio de 2021



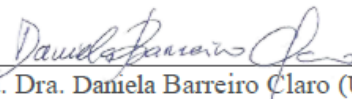
Prof. Dr. Frederico Araújo Durão (Orientador - PGCOMP/UFBA)




Prof. Dr. Carlos André Guimarães Ferraz (UFPE)



Prof. Dr. Carlos Eduardo Santos Pires (UFCG)



Profª. Dra. Daniela Barreiro Claro (UFBA)



Profª. Dra. Vaninha Vieira dos Santos (UFBA)

Dedico esta tese a minha mãe, que praticamente sozinha, me criou, educou e acreditou que eu seria capaz de chegar até aqui. Muito obrigado por todo amor e carinho.

ACKNOWLEDGEMENTS

Far beyond the academic work, the doctorate graduation represents the end of a cycle in my life. It is a significant cycle that impacts my professional and personal life. Perhaps because it had such an impact, it was also so hard to complete. In addition to the academic challenges, four years is a long period in which life also challenges you. Health issues, the pandemic, and other personal challenges happened in that period when I was trying to achieve my best academically. It was a challenge to stay focused during the troubling days. I am stressing the difficulties so the reader can truly understand how the support provided by the people I will quote contributed to this thesis.

First, I would like to acknowledge my advisor Professor Frederico Araujo Durão, for the opportunity of studying at UFBA and for guiding me through the postgraduate program. He always found time for discussing the research, reading my texts, and giving me good advice. Also, I am grateful for his comprehension and compassion during the hard days. Thanks for transforming me into a better researcher.

I express my gratitude to my colleagues at WISER who actively participated in the thesis development and my academic life. Together we had traveled, participated in congresses, and studied a lot. My thanks to Diogo, Gabriela, Paulo, Diego, and Amanda for this incredible experience that you have provided to me. I would also like to thank the other colleagues in the program whom I shared classes, especially Thiago Cerqueira and Augusto Coutinho. I could not have accomplished the postgraduate classes without the support of all my colleagues. They helped me to study and to relax in times of stress.

I also would like to acknowledge the Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) for financing the project. Special thanks to the CIDACS/Bahia specialists, especially Robespierre Pita, Elzo Pereira, and Marcos Barreto, for working with us during the development of the COVID-19 Geo-monitor. Also, thanks to Flaticon.com for providing an open database of icons and images that I used to design the figures.

I dedicate a special acknowledgment to my mother. Eliete Dias de Almeida believed in the power of education. She never spared any effort to guarantee me the best education possible. She encouraged me to read from an early age. My mother is the reason I grew up fascinated by science because she shaped me to seek knowledge. For this, I will be forever grateful. I love you, mom.

My thanks are extended to my wife Andreza Camilo for putting up with me all this time. You also fought with me, encourage me, and gave me strength. For that, I also will be forever grateful. A loving thought goes out to Sônia Freitas de Almeida Araújo and Hércules César de Araújo, who sheltered me in Salvador and took care of me. I also would like to thank my friend Igor Bastos who gave me immeasurable support in this doctorate. He helped to solve technical issues in my software and even reviewed long texts. Thank you very much!

"Even the smallest person can change the course of the future."

—J. R. R. TOLKIEN

RESUMO

Estima-se que 80% de todos os aplicativos baseados em dados possuem dados geo-referenciados. Consultas espaciais são amplamente empregadas para recuperar este tipo de dado eficientemente. Entretanto, o usuário tem um papel importante no processo de recuperação dos dados geo-referenciados. Um problema frequente neste cenário é quando o usuário não consegue descrever aquilo que ele deseja encontrar, dificultando a busca pelo ponto de interesse (POI) que o melhor satisfaça. Por décadas, pesquisadores propuseram técnicas para auxiliar usuários a expressar as suas necessidades. Dentre estas técnicas, pode-se citar os modelos booleanos, correspondência de padrões e expansão de consulta. Apesar da existência de alternativas importantes, faltam soluções que auxiliem o/a usuário/a a utilizar consultas do tipo preferencial que utilizem palavras-chave. O top-k Spatial Keyword Preference Query (SKPQ) é uma consulta deste tipo que surge como uma solução potencial para auxiliar usuários a encontrar POIs. O SKPQ seleciona POIs considerando a descrição de locais na vizinhança. Em essência, o usuário define uma restrição espacial (i.e. raio) e textual (i.e. palavras-chave) a ser satisfeita. Nesse contexto, esta tese propõe estratégias para melhorar a recuperação de informação proporcionada pela SKPQ e consultas similares. A contribuição desta tese pode ser dividida em três etapas. Na primeira, dois repositórios Linked Open Data (LOD) são explorados para melhorar a descrição dos POIs e suas vizinhanças. A descrição do POI no LOD contém mais informação do que nos bancos de dados espaciais tradicionais, o que resulta em uma descrição mais detalhada. Na segunda etapa, os resultados da consulta são personalizados para apresentar os melhores POIs para o usuário nas primeiras posições do rank. Ao explorar comentários relacionados aos POIs, o sistema identifica o objeto que melhor satisfaz a usuária da consulta e reordena o rank de acordo com a preferência dela. Na terceira etapa, utilizamos uma função probabilística para descrever a preferência por POIs próximos um do outro. Esta função probabilística é incorporada à função de ranqueamento para que a busca também considere esta preferência. Por fim, avaliou-se separadamente cada estratégia proposta nesta tese. A primeira estratégia alcançou melhora de 20% no Normalized Discounted Cumulative Gain (NDCG) ao utilizar palavras-chave aleatórias. Assim como encontrou POIs onde não era possível encontrar com a SKPQ. A segunda estratégia adiciona melhora de 92% no NDCG. Enquanto, a terceira estratégia melhora a consistência do rank, alcançando aumento no coeficiente Tau de 52%. Os resultados alcançados foram obtidos através de experimentos offline, utilizando dados de usuários reais em bases de dados públicas.

Palavras-chave: SKPQ, consulta espacial, linked open data, avaliação de consulta, processamento de consulta, personalização de consulta

ABSTRACT

It's been asserted that 80% of all data business has some locational reference. Spatial queries are widely employed to manipulate spatial data more efficiently. However, the user has a crucial role in the spatial information retrieval process when querying the needed information. A frequent problem in this context occurs when a user is unable to describe the object he/she desires to find. This problem hinders the search for the best point of interest (POI) to satisfies the user. For decades, researchers have proposed techniques to aid users in express their information need, such as Boolean models, pattern matching operators, and query expansion. Despite the existence of relevant alternatives in the field, there is still a lack of solutions to aid users of keyword preference queries to express their needs. The Spatial Keyword Preference Query (SKPQ) arises as a potential solution to assist users in finding POIs. SKPQ selects POI based on the description of features in their neighborhood. In essence, the user defines a spatial (i.e. radius) and textual (i.e. query keywords) constraint to be satisfied. In this context, this thesis aims at proposing strategies to improve SKPQ results. The contribution is threefold. First, two Linked Open Data (LOD) repositories (i.e. DBpedia and LinkedGeoData) are exploited to improve the features description. The feature description in LOD contains more information than traditional spatial databases, leading to a more detailed description. Second, the query results are personalized to present the best POIs for the underlying user. By exploiting reviews on POIs, the system identifies the object that best satisfies the user and re-order the rank with respect to the user preference. Third, we model the user preference in visiting locations near to each other using a probabilistic function. This function is incorporated into the ranking function to retrieve POIs considering this user preference. We evaluate each technique employed in this proposal separately. The first technique achieves a relative Normalized Discounted Cumulative Gain (NDCG) improvement of 20% when using random query keywords. Also, it finds POIs where SKPQ is unable to find. The second technique further improves the relative NDCG by 92%. Finally, the third technique improves the rank consistency achieving a Tau performance of 52%. The results achieved were obtained through offline experiments, using data from real users in public databases.

Keywords: SKPQ, spatial query, linked open data, query evaluation, query processing, query personalization

CONTENTS

I Overview

Chapter 1—Introduction	3
1.1 Motivation	6
1.1.1 Textual Description enhancement	6
1.1.2 Query results personalization	7
1.1.3 Probabilistic-based function to model average user preference	8
1.2 Problem Statement	9
1.3 Goal	9
1.3.1 Specific Objectives	9
1.3.2 Research Questions	10
1.4 Research Methodology	11
1.5 Statement of the Contributions	12
1.6 Thesis Structure	13

II Theoretical Background

Chapter 2—Spatial Information Retrieval	17
2.1 Spatial Information Retrieval Systems	18
2.1.1 Spatial Objects	19
2.2 Query	21
2.3 Textual Queries	22
2.3.1 Pre-processing	22
2.3.2 Similarity Measure	24
2.4 Query Personalization	25
2.5 Spatial Queries	26
2.5.1 Spatial selections	27
2.5.2 Spatial Indexes	28
2.6 Preference Queries	30
2.7 Spatial Preference Queries	32
2.8 Top-k Spatial Keyword Query	33
2.8.1 Spatio-textual Indexes	34
2.9 Top-k Spatial Keyword Preference Query	35
2.10 Summary	36

Chapter 3—The Semantic Web	39
3.1 Resource Description Framework - RDF	40
3.2 Ontologies	41
3.3 Linked Open Data - LOD	44
3.3.1 DBpedia	46
3.4 SPARQL	47
3.5 Summary	48

III Exploiting Open Data for Improving Spatial Keyword Query Applications

Chapter 4—Related Work	51
4.1 Description Enhancement using LOD	51
4.2 Personalization of Spatial Queries	54
4.3 Probabilistic Functions in Query Processing	56
4.3.1 Rank Based on probabilistic functions	57
4.3.2 Ranking on Top- k Spatial Preference Queries	58
4.3.3 Out of scope: Pareto Curve	60
4.4 Summary	61

Chapter 5—The Framework for Improving Spatial Keyword Query Applications	63
5.1 Modules Overview	63
5.2 Feature Description Enhancement Algorithm (SKPQ-LD)	64
5.3 Query Result Personalization Algorithm (P-SKPQ)	66
5.4 Probability-based ranking function	68
5.4.1 Data Analysis	68
5.4.2 The Ranking Function	69
5.4.2.1 Textual relevance (θ)	70
5.4.2.2 Pareto probability (Pr)	71
5.4.3 Probability-Based Search Model (PSM)	71
5.4.4 Probability-Based Ranking Re-Order (PRR)	72
5.5 Summary	74

IV Evaluation

Chapter 6—Experimental Evaluation	77
6.1 Module 1 Evaluation: Features' description enhancement	78
6.1.1 Experiment Setup	78
6.1.1.1 SPARQL queries	79
6.1.2 Datasets	79

6.1.2.1	Ground-truth Dataset for Experiment 1	80
6.1.2.2	Ground-truth dataset for Experiment 2	81
6.1.3	Metrics	81
6.1.4	Experiment 1: Evaluating Query Results	82
6.1.5	Experiment 2: Evaluating Feature Selection	85
6.1.6	Discussion: Datasets Characteristics and Features Description . .	86
6.1.7	Limitations and Points of Improvements	88
6.2	Module 2 Evaluation: Query Personalization	89
6.2.1	Experiment Setup	89
6.2.2	Datasets	89
6.2.3	Metrics	90
6.2.4	User profiles	90
6.2.5	Classification model	91
6.2.6	Choosing the Classifier	92
6.2.7	Results	94
6.2.8	Discussion	96
6.2.9	Limitations and Points of improvements	97
6.3	Module 3 Evaluation: Probability-based ranking function	98
6.3.1	Experiment Setup	98
6.3.1.1	Baseline Methods	98
6.3.2	Datasets	99
6.3.3	Metrics	101
6.3.4	Setting the Alpha Value	101
6.3.5	Experiment 1: Average User Satisfaction	102
6.3.6	Experiment 2: Varying the Number of Keywords	104
6.3.7	Discussion	106
6.3.8	Limitations and Points of Improvements	107
6.4	The COVID-19 Geo-monitor Use Case	108
6.4.1	Ranking POI considering patients with COVID-19 in their neigh- borhood	108
6.4.2	Presenting the nearest UBS to a patient	110
6.5	Evaluation of The COVID-19 Geo-monitor Use Case	111
6.5.1	Experiment Setup	113
6.5.2	Datasets	113
6.5.3	Metric	114
6.5.4	Results	114
6.5.5	Discussion	116
6.5.6	Limitations and Points of improvements	117
6.6	Summary	118

V Final Remarks

Chapter 7—Final Remarks

7.1 Contributions	121
7.2 Impressions	122
7.3 Future work	124
7.4 Dissemination	126
Appendix A—User Profile based on Room Aspect Value	141
Appendix B—Ratings and queries related to the aspect rating Room	147
Appendix C—Survey submitted to collect opinions from specialists about the prototype	155
Appendix D—Anonymized responses from specialists about the prototype	159

LIST OF FIGURES

1.1	Spatial area containing four different POIs and their respective textual descriptions.	4
1.2	Points of interest and features associated with textual descriptions.	6
1.3	Query execution for two different users over points of interest and features associated with their textual descriptions.	7
1.4	Common words in user review are used to personalize the query result.	7
1.5	Same feature in the neighborhood of distinct points of interest.	8
1.6	Schematic overview of the thesis structure.	13
2.1	Components of a Spatial Information Retrieval system.	18
2.2	Basic spatial objects.	20
2.3	AroundMe interface displaying the user location employed to search for locations around the user.	21
2.4	Textual database Keeper.	22
2.5	Inverted File example using existing terms in textual database Keeper.	23
2.6	Textual query execution example using an Inverted File.	25
2.7	Examples of the spatial selections nearest neighbor and range.	27
2.8	R-tree examples.	28
2.9	AR-tree example.	30
2.10	Spatial Preference queries examples using different ways to define the spatial neighborhood of a POI.	33
2.11	Spatial area containing bars and pubs.	34
2.12	Spatial Inverted Index.	35
2.13	POIs and features associated with their textual descriptions.	36
3.1	An example of an RDF graph describing the city of Salvador.	40
3.2	An ontology representing concepts and relationships between concepts.	42
3.3	Five Star Scheme for Linked Open Data.	45
3.4	Cross-domain LOD subcloud.	47
4.1	Timeline of related works on description enhancement of POIs.	52
4.2	Timeline of related works on personalization of spatial queries.	54
4.3	Timeline of related works on probabilistic functions to model the user preference.	58
4.4	Timeline of related works on ranking functions in top- k spatial preference queries.	59
5.1	Overview of our approach to automatically improve query results.	64

5.2	Overview of the textual enhancement algorithm by exploiting LOD. . . .	65
5.3	Overview of the personalization algorithm.	67
5.4	The distance distribution between the POI location and features in its spatial neighborhood.	69
5.5	Overview of the PSM algorithm.	72
5.6	Overview of the PRR algorithm.	73
6.1	Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size.	82
6.2	Results obtained with RQ and RQ-LD.	84
6.3	Relative NDCG improvements.	84
6.4	SKPQ-LD evaluation using OpinRank.	86
6.5	Classification model to learn user preference based on his/her past reviews.	91
6.6	Classifiers' performance evaluation varying training data and algorithms.	92
6.7	Measuring the agreement between the results generated by the classifiers with the expected ones.	94
6.8	Results obtained by P-SKPQ compared to SKPQ-LD, Fuzzy, and JW using Dubai dataset.	95
6.9	Results obtained by P-SKPQ compared to SKPQ-LD, Fuzzy, and JW using the London dataset.	96
6.10	Example of POI neighborhood considering the influence score.	99
6.11	POIs distribution in real datasets.	100
6.12	NDCG and Tau values achieved by PSM and PRR w.r.t different α values and different datasets.	102
6.13	NDCG varying the rank size k and the datasets.	103
6.14	Tau varying the rank size k and the datasets.	104
6.15	NDCG varying the number of keywords.	105
6.16	Tau varying the number of keywords.	106
6.17	Query parameters to search for places with the most number of COVID-19 infections in its neighborhood.	110
6.18	Query result presented by the application in a map and a table.	110
6.19	Detailed view of a specific POI in the query result.	111
6.20	List of patients in the POI's neighborhood and its distance to each patient.	112
6.21	Overview of our approach to automatically improve query results.	112
6.22	Example of linear scale.	114
6.23	5-point linear scale response of specialists about the use case functions.	115
6.24	Response of specialists about the textual description enhancement of POIs.	115
6.25	Response of specialists about their expectation about the prototype.	116

LIST OF TABLES

2.1	Dataset example with hotels.	31
4.1	Characteristics employed by different approaches to enrich textual descriptions.	54
4.2	Features comparison between our approach (P-SKPQ) and other Spatial Information Retrieval systems.	57
4.3	Comparison between features in Spatial Retrieval Systems that employ probabilistic rank functions and our novel probabilistic rank function.	59
4.4	Comparison between features in Spatial queries and our novel probabilistic rank function.	60
6.1	Characteristics of the Dubai dataset obtained from Mapzen.	80
6.2	Example of information available in OpinRank dataset related to the query “great location”.	81
6.3	Score of object p in traditional SKPQ compared with the score generated by SKPQ-LD, using hotels from Venice.	87
6.4	Score of object p in traditional SKPQ compared with the score generated by SKPQ-LD, using hotels from São Paulo.	87
6.5	Example of the user profile related to service aspect rating.	91
6.6	Experiment setting. Default values in bold.	98
6.7	Datasets characteristics.	100
6.8	Participants characteristics.	113

LIST OF ACRONYMS

IR	Information Retrieval	17
k	Number of results	98
LBS	Location-Based Services	54
LOD	Linked Open Data	51
MAP	Mean Average Precision	81
NDCG	Normalized Discounted Cumulative Gain	81
OSM	OpenStreetMap	100
P	POIs dataset	69
P-SKPQ	Personalized SKPQ	94
POI	Point of Interest	21
PRR	Probability-based Ranking Re-order	68
PSM	Probability-based Search Model	68
RDF	Resource Description Framework	40
REV	set of reviews	67
S2I	Spatial Inverted Index	34
SPARQL	Simple Protocol and RDF Query Language	39
SKQ	top-k Spatial Keyword Query	35
SKPQ	top-k Spatial Keyword Preference Query	35
SKPQ-LD	Spatial Keyword Preference Query with LOD	65
TAU	Kendall's Tau Coefficient	101
URI	Uniform Resource Identifier	40
VSM	Vector Space Model	6

PART I

OVERVIEW

INTRODUCTION

The popularization of social networks, mobile applications, and online services contribute to the growth of data available online. Within the overwhelming amount of data available online, there is a type of data called spatial data. This type of data is information about a physical object that can be represented by numerical values in a geographic coordinate system, like latitude and longitude (RIGAUX; SCHOLL; VOISARD, 2002). For example, the Eiffel Tower is described in Wikipedia by the latitude 48.8582 and longitude 2.2945. Spatial data is critical in a large number of application domains like information retrieval, transportation plan, or emergency response. In fact, 80% of all business data has spatial reference (DANGERMOND, 2017). Some search techniques retrieve spatial data in which the user has an interest, also known as Point of Interest (POI).

Location-Based Services (LBS) enable their users to describe, rate, and interact with urban spaces. These services manage spatial data to satisfy the user's desire. In order to help users to describe their information need, LBS provide a wide range of interfaces including eye-tracking, live location, GPS navigation on online maps, and the most traditional keyword searching. In Google Maps, for instance, users can use both keyword queries and map navigation to find the information they want. In OpenStreetMap, users can enter a particular coordinate to explore sights around. In essence, queries can provide flexibility to describe the characteristics of the retrieved data. The preference query is a popular query type that provides a manageable ordered set of answers, also known as rank. The rank is a useful tool because it filters and sorts the best answers from a large set of possible answers. Therefore, the rank avoids presenting an overwhelming amount of information to the user.

Preference queries can specify the user's desire using query keywords. For instance, a user looking for a Japanese restaurant can describe his/her preference with the query keywords "japanese restaurant". The relevant results for this query are the best matches between the user queries and the POI's description (CAO et al., 2012; CONG; JENSEN; WU, 2009). Under those circumstances, the more words in common, the better the object satisfies the user's needs. However, this evaluation method has limitations, especially



Figure 1.1: Spatial area containing four different POIs and their respective textual descriptions.

to objects with short textual descriptions. For example, suppose a spatial area (e.g. a city) containing two POIs like described in Figure 1.1. The query keywords are “japanese restaurant”, and each object has its own textual description “Oriental food”, “Asian Culinary House”, “Walmart Supermarket”, and “Samurai food”. This query will not return any results as there is no keyword matching, neither the word “japanese” nor “restaurant” appears in any textual description.

To tackle this problem, Xu and Croft (2017) tried to expand the query defined by the user to better describe his/her need, Thompson et al. (2017) proposed a method to include metadata to POIs to improve their description, while Lee, Lee and Hwang (2017) described a framework that enables voluntary users to generate or update POIs descriptions. Our proposal is to extend the limited descriptions of POIs using Linked Open Data (LOD) from the web. LOD is a set of practices for publishing and connecting structured data that enables a user to start browsing in one data source and then navigate to related ones through links (BERNERS-LEE, 2006).

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). In essence, researchers employ LOD to combine information from different data sources about an object. 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. One basic query ranking function is based on the venue’s popularity which can easily be estimated by ratings (i.e. items rated 5 are top-ranked). For example, Google Maps users assign rates (e.g. stars) to places they have visited, indicating the place quality. The query

employs this information to generate a rank that satisfies the user’s need and is ordered by the places’ rating. This basic ranking function produces the same rank for two users in the same location, although they are completely different individuals. In fact, generic spatial information retrieval is useful for taking a glance at the most popular places in a region. However, the user’s decision-making process considers many facets in general, such as the recommendation of an item by a friend (GASPARETTI, 2017) and the user’s preference about the object (KWON; SHIN, 2008).

Query personalization provides tailored content to individuals based on knowledge about their preferences and behavior (HAGEN; MANNING; SOUZA, 1999). Typically, the personalization filters objects with a low value to the user or sorts them to present high-value data first. A user profile containing user interests is the primary tool exploited to personalize queries. Several works in the literature study query personalization (KWON; SHIN, 2008; MARGARIS; VASSILAKIS; GEORGIADIS, 2018; RATHOD; DESMUKH, 2017). These works make use of user preferences (personal or collaborative) that are stored in preference repositories.

Another possibility is the distance between the places in the search space to define the items’ position in the query result. Several top- k queries search for POIs analyzing the features in the vicinity of these points (CAI et al., 2019; ANDRADE; ROCHA-JUNIOR, 2019; TSATSANIFOS; VLACHOU, 2015). These queries often employ a ranking function considering the user’s preference with another constraint, such as textual relevance of a feature to a query (TSATSANIFOS; VLACHOU, 2015), or the distance between the feature and the Point of Interest (POI) (YIU et al., 2007; YIU et al., 2010).

Statistical analysis suggests that users prefer to visit POIs that are close to each other. Researchers have recently discovered a spatial clustering pattern in human mobility behavior, demonstrating the effective use of this pattern in POI recommendation (LIAN et al., 2014; YE et al., 2011; ZHANG et al., 2018). Inspired by these findings, some researchers also have adopted probabilistic functions to model the user mobility pattern (FENG et al., 2017; ZHANG et al., 2018; ZHU et al., 2015). Therefore, this research investigates the use of a probabilistic function to model the user preference to visit POIs close to each other. The ranking function is combined with the probabilistic function to search for POIs and re-order the query result.

In this D.Sc. thesis, we focus on the results presented to the user by spatial preference queries. We propose a framework that exploits LOD to improve the limited description of POIs, query personalization to re-order the query result considering the user preference, and probabilistic functions as another possibility to re-order the query results. The next section details the topics that motivate the research. We start by illustrating with a detailed example of textual description limitation of POIs in Section 1.1.1. Then, we describe an example of query personalization and its benefits in Section 1.1.2. Section 1.1.3 concludes by explaining the use of a probabilistic-based function to identify POIs in the search space that satisfies the user the most.

1.1 MOTIVATION

1.1.1 Textual Description enhancement

Several queries are processed using the Vector Space Model (VSM) to evaluate the textual relevance between query keywords and object’s textual description (CAO et al., 2012; CONG; JENSEN; WU, 2009; ALMEIDA; ROCHA-JUNIOR, 2016). The VSM indicates that two strings are textually 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 POIs based on spatio-textual objects¹ of reference (features) in their spatial neighborhood.

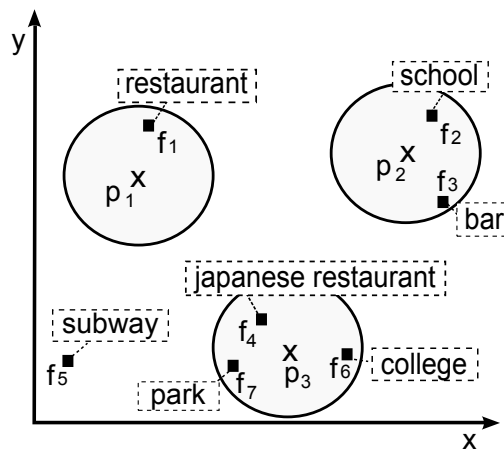


Figure 1.2: Points of interest (p) and features (f) associated with textual descriptions.

For example, Figure 1.2 describes a spatial area with POIs 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 criterion (represented by the circle around the points p). The SKPQ returns the point 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 criterion. Further details about SKPQ processing are presented in Section 2.9.

Now, suppose a SKPQ with query keywords “oriental food”. Considering Figure 1.2, this query does not return any objects. Neither the word “oriental” nor “food” is 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”. For this reason, we propose a solution using LOD repositories to enhance features’ textual description, adding more words to describe the object and possibly contributing to the query not neglect a useful object to the user. Thereby, a wider textual description for features f can improve their textual relevance with the query keywords. If object f_4 had a better textual description, the word “food” or “oriental” might appear in the

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

textual description. In this scenario, the semantic relationship offered by the LOD can also contribute to the feature description.

1.1.2 Query results personalization

Figure 1.3 highlights a generic SKPQ response that does not consider the user’s implicit preferences. Suppose that two users are looking for a hotel near a Japanese restaurant, but they have different opinions about the definition of a good hotel. Unlike User 1, User 2 prefers a comfortable hotel instead of a cheaper one. We can suppose that due to User 2 past reviews, in which he/she rated five 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 he/she has a preference for comfortable hotels.

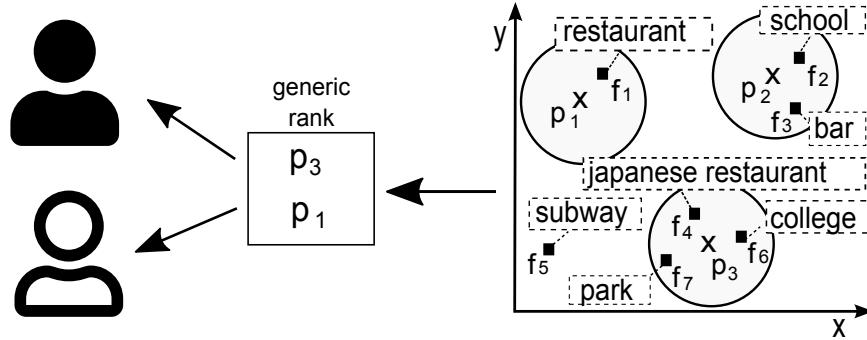


Figure 1.3: Query execution for two different users over points of interest (p) and features (f) associated with their textual descriptions.

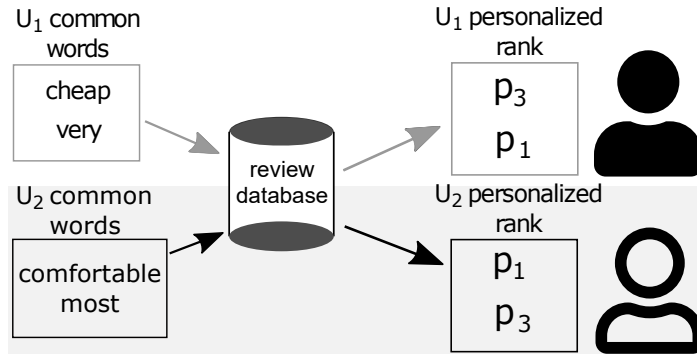


Figure 1.4: Common words in user (U_x) review are used to personalize the query result.

Using textual classifiers, we can identify which words are common in the user’s past reviews. These words describe the implicit preference of the user, so we use them to personalize the SKPQ. Thereby, we model the user preference using his/her past reviews. The review database is employed to describe an unknown POI to the user through reviews from different users, enabling the system to modify the POI’s position in the query rank accordingly to the user preference model. The POIs whose reviews are similar to the

positive reviews in the user model receive a boost in the ranking position (or decrease whether the reviews are negative), as described in Figure 1.4. Together with the feature description enhancement, we explore personalization aiming for an improvement in the accuracy of the SKPQ results.

1.1.3 Probabilistic-based function to model average user preference

Statistical analysis suggests that users prefer to visit points that are close to each other, resulting in a spatial clustering pattern. Researchers have applied this pattern successfully in the recommendation of points of interest. To tackle this problem, we model the clustering pattern in a novel probabilistic score function that considers the distance distribution in the point’s neighborhood. In essence, the score function measures the user preference for a point considering the point’s distance to its features. The proposed score function can be adopted by any spatial preference query.

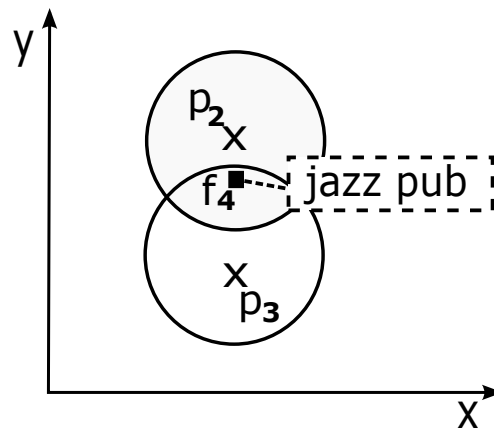


Figure 1.5: Same feature in the neighborhood of distinct points of interest.

The novel score function is relevant to distinguish points of interest when the textual relevance is not enough. For example, consider a user interested in finding a hotel near a jazz pub. Using a SKPQ, the user specifies the query keywords “jazz pub” and the spatial selection criterion. Figure 1.5 illustrates a spatial area where one feature f_4 shares the spatial neighborhood of the points p_2 and p_3 . Since the score of p depends only on the textual relevance between the query keywords and the feature’s description, points p_2 and p_3 receive the same score. However, p_2 is closer to the feature than p_3 . Considering the statement that users prefer to visit points that are close to each other, we can suppose that the average user is better satisfied with p_2 rather than p_3 . Hence, we propose a novel score function based on a Probabilistic distribution to describe and measure the user preference for features close to the point of interest. Adopting our function, the query considers both the textual relevance and the average user spatial preference. Thereby, the point p_2 goes to the top of the rank in the scenario described in Figure 1.5.

1.2 PROBLEM STATEMENT

The problem addressed in this thesis is the incapability of single spatial keyword querying to satisfy user needs by neglecting important and available information. Usually, these queries manipulate objects with short textual description, hindering the query capability to identify the objects that satisfy the user information need. Moreover, traditional keyword queries only consider the user information need depicted by query keywords. Overlooking that users have personal preferences even when they type the same query keywords. These are the major challenges faced in this research:

- Retrieve the point of interest that matches the user query when there is little or no information describing it.
- Accurately capture and apply the user interest from spatial information retrieval systems.
- Employ an average user preference to improve query performance, considering the distance of places in the search space and the user query keywords.

These topics are underlying support for our research questions and conduct the techniques that tackle each particular challenge. Thereby, we propose a location-based solution that exploits the benefits of LOD to enhance the textual description of POIs. LOD enables applications to navigate along with links into related data sources. By navigating these links, the approach obtains an enhanced textual description for the objects. Also, the query results are personalized to consider the user's unique preferences. A classifier exploits user reviews to define the user preference for each object in the query result. Likewise, the query result can be re-ordered considering an average user preference instead of the personalized one. An approach suitable to scenarios where there is no information available about the user.

1.3 GOAL

In this project, we aim to improve spatial keyword preference queries towards the retrieval of more accurate query results. This way, we propose a framework to support improvements in spatial information retrieval. Our solution combines query personalization or an average user preference with the object's description enhancement in order to increase query accuracy. In this way, the best item for the user appears first in the rank.

1.3.1 Specific Objectives

The following are the specific objectives pursued by this thesis:

- SO 1.** Conduct a literature review to identify the methods applied to improve spatial keyword preference queries.
- SO 2.** Designing and exploiting the adaptation of user queries to benefit from Linked Open Data, indicating the application scenarios where this query can be useful.

SO 3. Proposing algorithms to process the personalized SKPQ coupled with the textual description enhancement algorithm.

SO 4. Proposing algorithms to process a probability-based SKPQ coupled with the textual description enhancement algorithm.

SO 5. Developing a use case to apply queries that can benefit from Linked Open Data in a real case scenario.

1.3.2 Research Questions

The following research questions were proposed by considering the problems and objectives exposed:

RQ 1. How can we sort the best POIs retrieved by spatial preference queries to satisfy the user?

During the last years, new queries have been proposed (LE et al., 2019; QIAO et al., 2020; ZACHARATOU et al., 2019), and new indexes have been introduced to process these queries efficiently (HAN et al., 2016; ZHU et al., 2019). In contrast, some researches focus on the query result evaluation (KELES, 2018; SONG et al., 2017). Despite the existence of studies proposing techniques to enhance the results generated by spatial queries, it is still missing a study on queries from SKPQ class, in which the POI is also evaluated by considering features in its neighborhood. We investigate algorithms and strategies in related studies to develop a suitable solution to apply to SKPQ and related queries.

RQ 2. How to exploit Linked Open Data to process spatial preference queries?

Several studies adopt LOD repositories to improve textual descriptions of POIs (ALMENDROS-JIMÉNEZ; BECERRA-TERÓN; TORRES, 2019; KARAM; MELCHIORI, 2013). However, the use of improved textual descriptions by spatial keyword preference queries to boost query accuracy remains unexplored. In this thesis, we access different LOD repositories to concatenate distinct textual descriptions that describe the same POI. Thereby, we propose an approach that automatically provides textual descriptions for POIs, avoiding the need for human participation to validate the generated description.

RQ 3. How to model the user preference to improve SKPQ results?

Search engines do not always meet the expectations of their users. Particularly the ones that rely only on query keywords to filter the search space; because they do not take into account the user characteristics. Since different users might have distinct preferences, they also might have different expectations about the query result. User modeling provides models of the user preferences, abilities, and goals (DURAO, 2012). User modeling is widely applied in the personalization of Information Retrieval Systems (MARGARIS; VASSILAKIS; GEORGIADIS, 2018; XIA;

GONG; ZHU, 2011). Personalization concerns the construction of user profiles, or models, to provide personalized services. In this thesis we investigate the role of personalization to improve spatial keyword preference queries.

RQ 4. It is possible to combine different techniques to improve SKPQ results?

There are a plethora of techniques to improve user satisfaction regarding a search result. In this thesis, we explore the literature review to adapt existing methods or propose novel ones to retrieve the POIs from the search space in the best way to the user. Despite the existence of many solutions such as query personalization or user modeling (RATHOD; DESMUKH, 2017; ZHANG et al., 2018), there is a lack of experimentation of these techniques in spatial preference keyword queries. We exploit different techniques aiming to improve the accuracy and rank consistency of SKPQ and other similar queries. We also conduct experiments combining these techniques to evaluate further improvements.

RQ 5. How to evaluate the proposed approaches?

We investigate methods to objectively assess the query’s ability to meet the user needs. Volunteers can define the relevance judgments (correct answers) to measure the query performance regarding the accuracy, precision, and other related metrics. However, test collections are expensive to build, and the relevance judgments are the most expensive (HARMAN, 2011). Recent evaluation papers adopt metrics that reflect the user satisfaction with the query output (IOANNAKIS et al., 2017; SONG et al., 2017). Thereby, we conduct two types of experiments in this thesis: offline experiments that consider existing real-user judgments datasets coupled with metrics that measure the average reputation of the query output, and a user interview to collect feedback from real users regarding SKPQ and the techniques employed to improve the search.

1.4 RESEARCH METHODOLOGY

In this research, we employ a quantitative research methodology based on experimental evaluations and a survey to collect feedback from specialists (APUKE, 2017; OCHIENG, 2009; QUEIRÓS; FARIA; ALMEIDA, 2017). The experimental evaluation is a simulation in which we analyze the evidence collected from the results and test our hypothesis based on that evidence. In the experiments, we select one variable and study its effect on other variables. For most of the research questions, we perform a full evaluation of the hypothesis based on the proposals’ implementation on real data.

The survey is a popular technique that uses a set of questions to collect data from a person involved in the research (QUEIRÓS; FARIA; ALMEIDA, 2017). We presented our technique to specialists, submitted the questions and collected the responses. The responses were evaluated and discussed to identify limitations and points of improvement. The research protocol is composed of the following steps:

1. **Technology Research and Implementation** - Techniques to improve the SKPQ were identified by the knowledge obtained from the literature review. The Linked Open Data is explored to improve the description of POIs. SKPQ is employed in every research phase because it has characteristics of two popular queries like the preference query and keyword query. Thereby, a framework is proposed with five different modules to refine spatial keyword preference queries in different stages of query processing. It combines the enhanced object's description with query personalization and the probability-based rank function. Then, the modules of the framework are assessed to measure the query result accuracy.
2. **Experimental evaluation** - The framework modules are evaluated by accessing two real-world datasets, such as a review dataset obtained from TripAdvisor and user data extracted from Google Maps. We employ traditional metrics described by studies on IR evaluation to assess the query result quality. Under those circumstances, we conduct experiments considering keywords from real users. Thereby, we can understand the benefits of our approach in a realistic scenario. Each module has a unique methodology section explaining the unique characteristics of its experimental evaluation. During the evaluation, we select one variable to determine its effect on other variables. The main variables employed in the experiments are cardinality (i.e. the number of POIs stored in the dataset), the number of results expected in the query, and the number of keywords employed in the query.

1.5 STATEMENT OF THE CONTRIBUTIONS

In this research, the aim is to contribute to the spatial information retrieval research area in three ways. The contributions are described in sequence:

1. **Query processing algorithms** - Five new algorithms to improve the SKPQ result are proposed by analyzing the studies conducted in the research area. One algorithm to improve the textual description of POIs using Linked Open data; two algorithms to personalize queries using a textual classifier; and, two algorithms to search for POIs considering a probabilistic-based ranking function. All algorithms are novel approaches to process spatial preference keyword queries.
2. **Proposal evaluation** - In order to evaluate the framework modules, we conducted an experimental evaluation and compared the results obtained with baselines. 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. Additionally, the use case is employed to assess the query with real users who provided questionnaire-based feedback. We also discuss the key findings of our study, as well as outline directions for future research. The discussion contributes with evidence, data, and procedures that may support replication by other researchers interested in the spatial information retrieval field.
3. **Real Use Case Scenario** - The text enhancement improvement is applied to a use case scenario to monitor the COVID-19 outbreak. A Web application is developed

for users to pose an SKPQ and search for the nearest health unity. In this use case, the SKPQ search for POIs considering the number of COVID-19 cases in the neighborhood. Thereby, the query returns a rank of places where the disease spread the most, indicating dangerous locations in the city. The use case is evaluated by specialists in epidemiology and computer science.

1.6 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 project. In addition, Figure 1.6 depicts a schematic overview of the thesis structure. Thereby, the structure can be outlined in the following chapters:

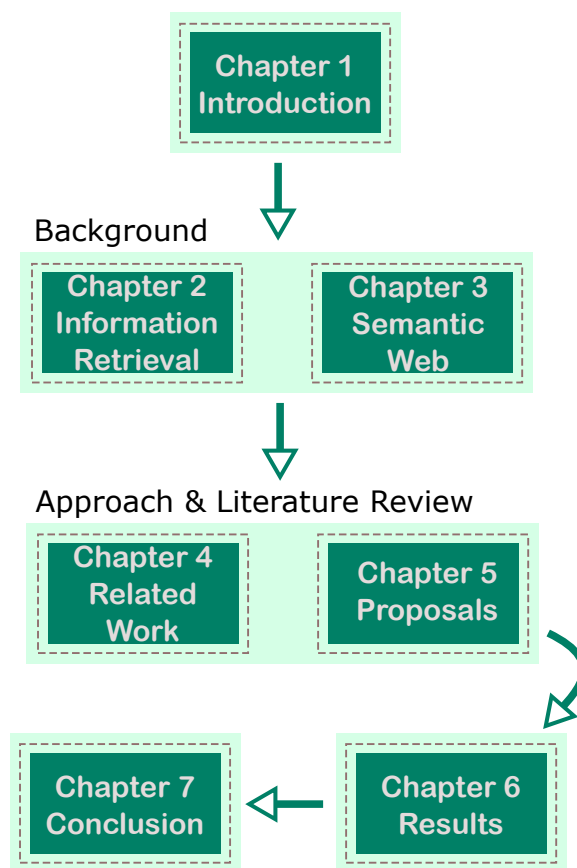


Figure 1.6: Schematic overview of the thesis structure.

- **Chapter 2** - presents an overview of the basic concepts that guide this thesis, such as spatial information systems, spatial keyword preference queries and their respective storage indexes. This chapter illustrates each concept with examples to facilitate understanding;

- **Chapter 3** - introduces the Resource Description Framework (RDF) and Linked Open Data. Also, we describe the Semantic Web and related concepts. The chapter also exemplifies each concept to facilitate understanding;
- **Chapter 5** - defines in detail the strategy to improve SKPQ and describes the algorithms of each framework module to improve it;
- **Chapter 4** - encompasses the literature review on textual description enhancement, query personalization, and probabilistic functions applied to spatial keyword preference queries rank functions;
- **Chapter 6** - describes the conducted experiment together with the methodologies, the datasets, and the results obtained. This chapter also explains how the datasets were obtained and their characteristics. Then, it details the experiment parameters and plots the results in graphs to graphically visualize the results;
- **Chapter 7** - concludes the thesis and discusses future work.

PART II

THEORETICAL BACKGROUND

SPATIAL INFORMATION RETRIEVAL

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

However, the popularization of GPS enabled devices increases the amount of unstructured data associated with spatial coordinates available for indexing and retrieval (PURVES et al., 2018). Spatial Information Retrieval (also known as Geographic Information Retrieval - GIR) is the IR field that seeks to develop spatially-aware systems, supporting queries that manipulate spatial coordinates (ADAMS, 2018). Thereby, these systems can efficiently store and retrieve data associated with spatial coordinates.

Spatial IR systems and services are popular today, helping millions of users worldwide to find information that satisfies their needs. Web search engines like Bing Maps¹ and Foursquare² are examples of spatial IR systems. They help find people or locations, exhibit user reviews about locations, support comparisons such as price comparisons between store products, or help calculate the distance from the user location to locations of interest (BAEZA-YATES; RIBEIRO-NETO, 2011).

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

¹<<https://www.bing.com/maps>>

²<www.foursquare.com>

2.1 SPATIAL INFORMATION RETRIEVAL SYSTEMS

A spatial IR system provides access, storage, and management to objects (e.g. hotels and restaurants) and 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 on the subject.

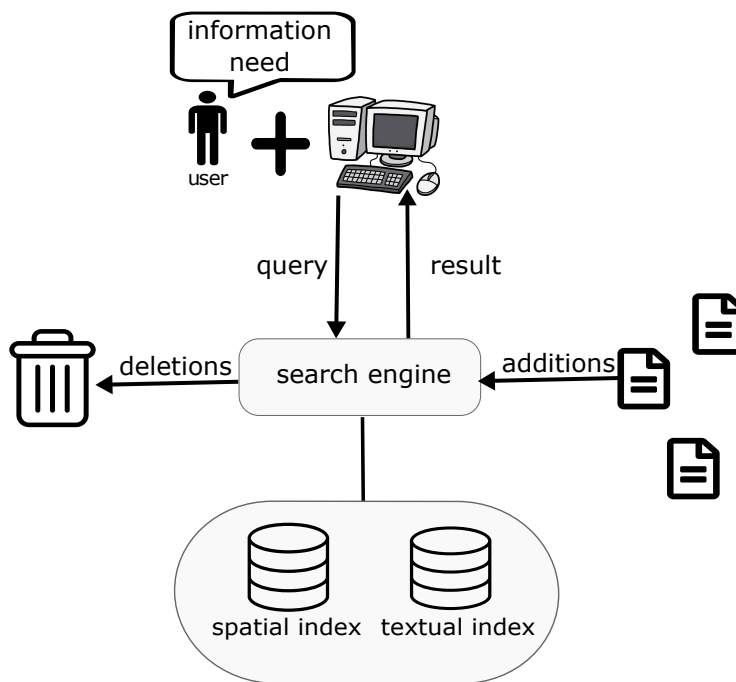


Figure 2.1: Components of a Spatial Information Retrieval system. Source: Adapted from Büttcher, Clarke and Cormack (2016).

Figure 2.1 illustrates the main components in a Spatial Information Retrieval system. Before searching, a user has an *information need* that drives the search process. Büttcher, Clarke and Cormack (2016) state that *information need* sometimes refers to a topic, particularly when it is presented in written form. In order to satisfy his/her information need, the user constructs and issues a query to the IR system. Usually in a Web search³, the query is composed of two or three terms (FINKELSTEIN et al., 2002; SUGIURA; ETZIONI, 2000). Frequently, the expression “term” is used 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 couple with query terms. For example, “retriev*” might match any wording starting with that prefix

³The word “search” frequently replaces “(information) retrieval”, therefore we use the two synonymously in this thesis (MANNING; RAGHAVAN; SCHÜTZE, 2010).

(e.g. retrieve, retrieval, retrieves).

The search engine processes the user’s query on the user’s local machine or a cluster of machines in a remote geographic location. For this purpose, it maintains and 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. To summarize, the inverted index provides the mapping between terms and locations in the collection in which they occur. The search engine uses these indexes for searching and ranking. The spatial indexes are further described at Section 2.5.2 and the hybrid indexes are described at Section 2.8.1. Because of the size and complexity of these indexes, efficient algorithms are necessary to access and update them.

The search engine maintains a collection of 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 to present the best document for the user first. Therefore, ranking functions use the statistics maintained by search engines to compute a document score that defines its position in the rank. Moreover, the search engine can report meaningful results using the original content of the documents (BÜTTCHER; CLARKE; CORMACK, 2016). For example, the search engine can use the places’ description to retrieve a pharmacy near the user location.

In summary, the spatial IR system employs one or many indexes, a collection of statistics, and other data; to process the user query and return a list of results. The search engine computes a score for each document to perform relevance ranking. Then, the system sorts the documents according to their scores and may remove redundant results. According to Büttcher, Clarke and Cormack (2016), a Web search engine might report only one or two results from a single host or domain, benefiting pages from different sources. The problem of scoring documents with respect to a user’s query is one of the most fundamental in the field (LUCCHESI et al., 2016; SKORKOVSKÁ, 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, 2002). This system provides additional support to spatial data⁴ 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 2.5.2 and 2.8.1.

In this research, the space of interest is the Euclidean space R^d , together with the Euclidean distance. Hence, the Euclidean space dimension d is 2. Thereby, this thesis focus on the region of R^2 that contains the relevant objects. This region is bounded by a sufficiently large rectangle parallel to the axes of the coordinate system for simplification. The region R^2 is defined as search space whenever a search operation is performed in its area. Points are elements of this space. A point has a pair of (Cartesian) coordinates that is denoted as x (the abscissa) and y (the ordinate) (BAEZA-YATES; RIBEIRO-NETO, 2011; MANOLOPOULOS; PAPADOPOULOS; VASSILAKOPOULOS, 2005).

⁴Spatial data is any information associated with geographic coordinates (e.g. latitude and longitude).

A geographic object has two components: (1) a description and (2) a spatial component, also referred to as *spatial object*, that 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 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 (RIGAUX; SCHOLL; VOISARD, 2002).

The spatial object does not correspond to any standard data type, such as string, integer, or float. 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 spatial database system offers support for modeling spatial data.

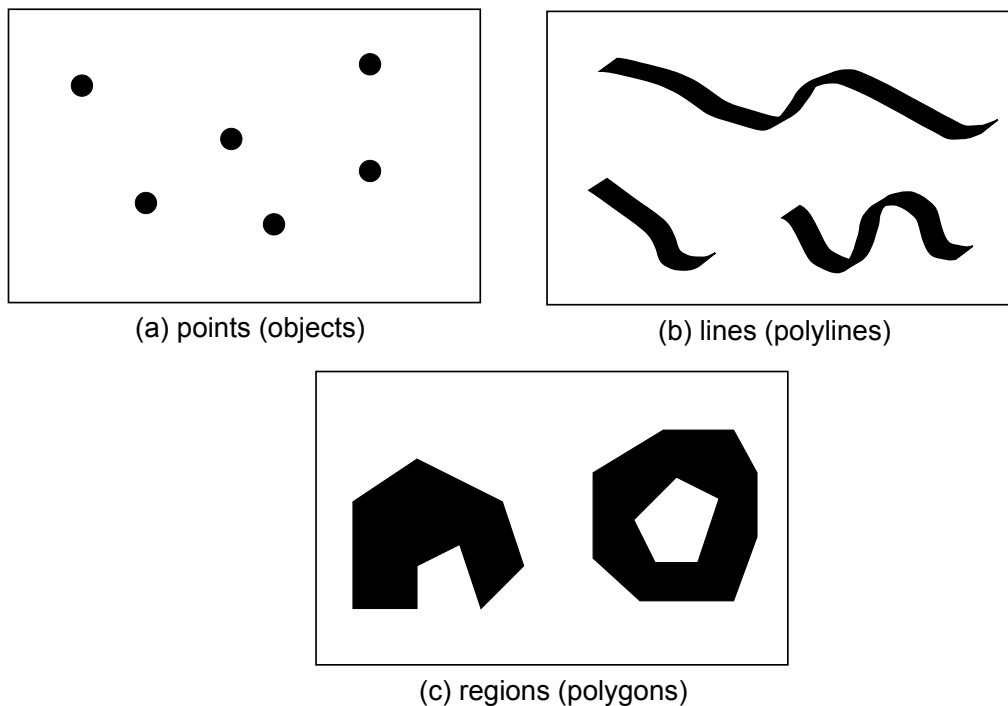


Figure 2.2: Basic spatial objects. Source: Adapted from Rocha-Junior (2012).

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 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 research, the user’s Point of Interest (POI) and other points (features) in the search space are “spatial objects” whose area is not relevant. The POI is a spatial object the user has an interest in, and a feature is a spatial object in the vicinity of a POI. In our scenario, all spatial objects are associated with a textual description. For this reason, we also use the term “spatio-textual object” as 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. Currently, popular applications such as Google Maps and Booking employ spatial queries to retrieve spatial objects based on the user’s query keywords or other object’s characteristics (e.g. cheapest hotel in Salvador). Spatial queries examples are described in Section 2.5 to Section 2.9. First, a definition of textual queries is presented because this thesis employs a query with textual and spatial characteristics (hybrid).

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 he/she can use keywords, spatial coordinates, and even voice commands to define his/her information need into a query (HEARST, 2011). Figure 2.3 illustrates the application AroundMe⁵ that receives the user coordinates (blue dot), processes a query, and returns the locations around the user (red pins).

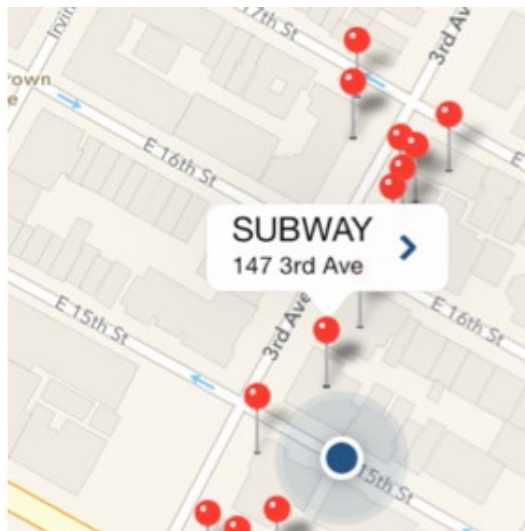


Figure 2.3: AroundMe interface displaying the user location (blue dot) employed to search for locations (red pins) around the user.

Although users typically issue simple queries, IR systems support complex Boolean and pattern matching operators. These operators can be used to limit a search to a particular web site, to specify constraints on fields such as author and title, or to apply

⁵<http://www.aroundmeapp.com/>

other filters that restrict 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 query operators 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 in finding the exact phrase inside a document. Under those circumstances, Boolean operators (AND, OR, and NOT) are used to combine a set of terms in a query description. For example, the user could define “Leonardo da Vinci” AND “sculptures” as his/her query keywords (ZOBEL; MOFFAT, 2006).

2.3 TEXTUAL QUERIES

The textual query is a key technology to search engines (BAEZA-YATES; RIBEIRO-NETO, 2011; ZOBEL; MOFFAT, 2006). A user can type one or more keywords in a textual query to describe the document he/she wants to retrieve (MANNING et al., 2012). 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 that employ textual queries.

A textual database is a collection of textual data like web pages, academic publications, or e-mails. Each element from a textual database is called *document*. Accordingly to Manning et al. (2012), *document* is any unit chosen to build a Information Retrieval System. In a typical IR System, a user describes the document he/she desires to retrieve 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 (2006).

For example, based on the textual database described in Figure 2.4, a user can identify a document he/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 extracting the terms from each document in

a textual database. For this purpose, a pre-processing stage using Natural Language Processing (NLP) named *parsing* is applied to the documents. Zobel and Moffat (2006) divides the parsing in three stages: *stem*, *casefold*, and *stop words* removal. In this thesis, the parsing is executed on the data used to process the SKPQ.

The *stem* consists of removing variations of the same word, keeping only the non-inflected verb. The following result is obtained by applying the *stem* in document 1: “*The old night keep keep the keep in the town*”. 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. *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*, it is possible to obtain the following terms from document 1: “*old night keeper keeps keep town*”. 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 one set of inverted lists (known as postings list). Moreover, each term t in a collection has a corresponding inverted list. This list contains an identifier (D_{id}) for each document (D^6) 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 that contain the term t and a pointer to the inverted list correspondent to the term t (ZOBEL; MOFFAT, 2006).

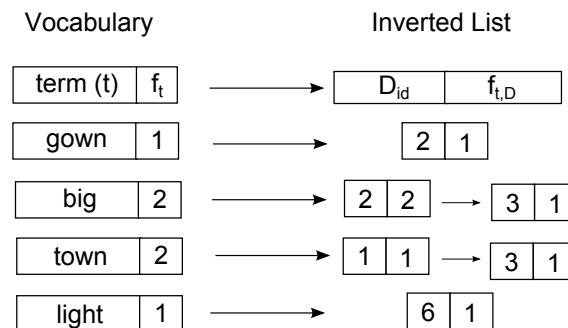


Figure 2.5: Inverted File example using existing terms in textual database *Keeper*.

For instance, Figure 2.5 illustrates 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 containing 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 (ALMEIDA, 2015).

⁶Each line in Figure 2.4 represents a document D while the text inside the line represents the textual description of D .

2.3.2 Similarity Measure

An Information Retrieval System that employs textual relevance to retrieve documents also uses a *ranking* function 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; 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 vector⁷ 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 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 textual 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., 2012; ROCHA-JUNIOR, 2012). In this thesis, we use the formulation proposed by Zobel and Moffat (2006), presented in Equation 2.1 which employs the metrics described early.

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

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 $w_{t,T} = \ln\left(1 + \frac{N}{f_t}\right)$. 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 (BAEZA-YATES; RIBEIRO-NETO, 2011; 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 using an Inverted File. Initially, each document has its textual score equals to zero, while a sum array A , of size N , is

⁷The weight vector is formed by the terms weight t in T .

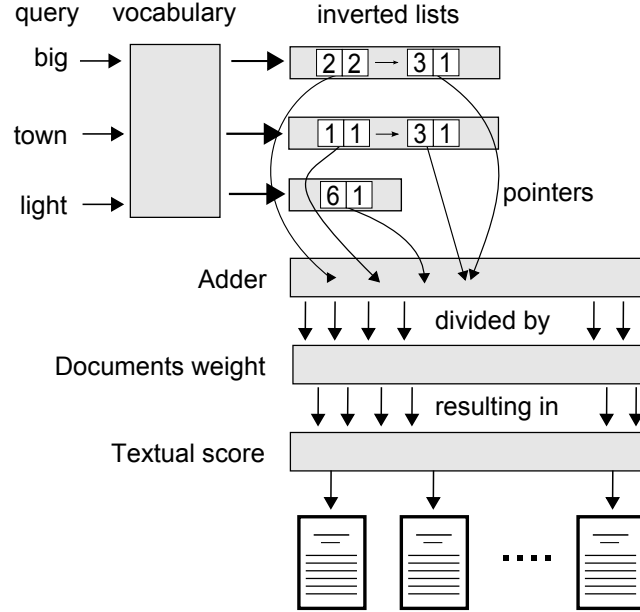


Figure 2.6: Textual query execution example using an Inverted File (IF). Source: Adapted from Zobel and Moffat (2006).

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), $W_D = \sqrt{\sum_{t \in D} (w_{t,D})^2 \cdot \sum_{t \in T} (w_{t,T})^2}$ (from Equation 2.1). Last, the documents are ordered by their respective textual scores and presented to the user.

2.4 QUERY PERSONALIZATION

Query personalization refers to IR systems that consider a user model to rank the search results. The user model represents the interests of a particular user. Thereby, query personalization respects the interests of an individual user when retrieving relevant results. The user’s interest can be determined without user participation (implicit interaction) or requiring active user feedback (explicit interaction). For example, Rathod and Desmukh (2017) presents a search engine that adopts explicit user interaction; since it asks the user to rate results that he/she judges relevant manually. In contrast, our proposal employs the implicit user interaction since it does not ask the user to provide any information during the search process (see Section 5.3 for further detail).

A common problem in the IR research area is that users often have different intentions when issuing a query (XU et al., 2008). It is hard for users to define precise queries that describe their information need (QIU; CHO, 2006). Consequently, the query results

do not match the user's intention. Under those circumstances, personalization rises as a promising solution to overcome this problem by providing methods to learn user's preferences and rank the search results accordingly. In this project, we investigate this problem and propose to learn the user preferences by training a classification model with users' past reviews on items of their interest. Then, the classification model is employed by the framework module to personalize the search, re-ordering the query result based on the user's interest.

Text analytics has become increasingly popular because text data is in abundance on the Web, frequently appearing on social networks, emails, and chat sites (AGGARWAL, 2018). We apply machine learning to extract useful insights from users' past reviews about items of their interest. Machine learning algorithms generate a classification model based on labeled documents (training data). The classification model, also known as a classifier, can predict the label of new documents. It is important to notice that the classifier performance is directly related to the training data quality and the representation model quality (AGGARWAL, 2018; JIN et al., 2016; SINOARA et al., 2019).

Classify textual data requires a structured representation of unstructured data. This structured representation has to maintain the patterns used by machine learning algorithms to generate the classifier. This way, the structured representation of texts is an open challenge for the text mining research community (SINOARA et al., 2019). According to Sinoara et al. (2019), the most popular text representation model is the Vector Space Model (VSM), where each document is represented by a vector whose dimensions correspond to features found in the text corpus. When features are single words, the text representation is called bag-of-words (BOW). The BOW representation is based on independent words and does not express word relationships, text syntax, or semantics (SINOARA et al., 2019). It is a simple text representation model that is easy to construct and has been provided satisfactory results in several applications.

2.5 SPATIAL QUERIES

Databases adapted themselves to store and organize different types of data efficiently. The spatial data availability associated with technological advancement made possible a scenario where spatial data is the core of many applications (MANOLOPOULOS; PAPADOPOULOS; VASSILAKOPOULOS, 2005; RIGAUX; SCHOLL; VOISARD, 2002). Today, any individual using a smartphone is a potential spatial data provider due to the Global Positioning System (GPS) popularization.

Satellite images, medical equipment, or Geographic Information System (GIS) are other spatial data sources in addition to GPS that 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 it is required to analyze the data in detail. Thereby, spatial queries are essential to analyze and obtain useful insights about spatial data efficiently.

2.5.1 Spatial selections

Because of the large volume of spatial data available for search, the spatial selection named “range” is employed frequently to filter the relevant data. The range uses its predicate (radius) to filter the data around the desired location. The spatial selections based on predicates are among the most significant spatial query types employed by spatial IR systems (GÜTING, 1994; ROCHA-JUNIOR, 2012). Given a database, a spatial selection returns the set of objects that 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.

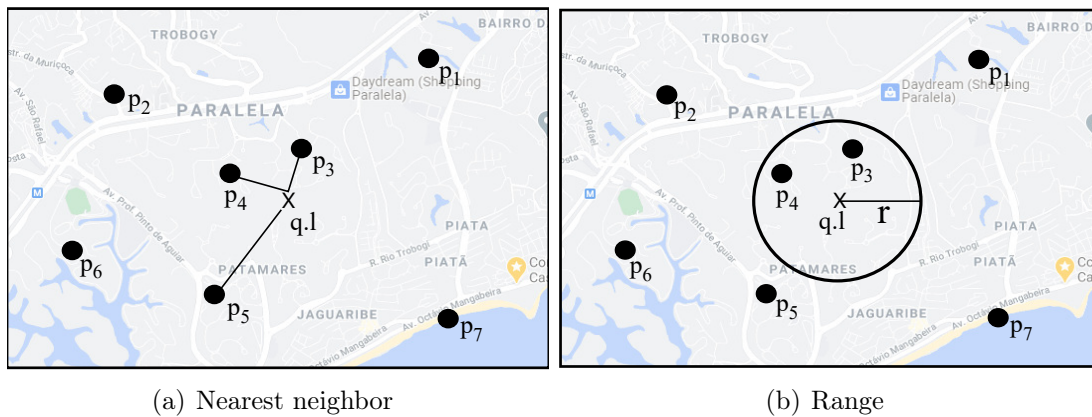


Figure 2.7: Examples of the spatial selections nearest neighbor and range.

The nearest neighbor (k-NN) is one example of a spatial query. The sentence “*find the three nearest restaurants from my actual location*” represents the k-NN. Therefore, only the the three objects with the shortest distance to the user location are selected by the k-NN spatial query. Given a set of POIs P ($p \in P$), the query location $q.l$, and an integer value k , the query retrieves the k POIs with the smallest distance to $q.l$ (shortest distance $dist(p, q.l)$ between any p and $q.l$. In other words, $\forall p' : dist(p, q.l) < dist(p', q.l) | p, p' \in P, p \neq p'$) (ROCHA-JUNIOR, 2012; YIU et al., 2007). Figure 2.7(a) illustrates a search space with seven POIs p and a query location $q.l$. Consider a user interested in finding the three nearest POIs (3-NN) from his/her query location. Thereby, the POIs p_3 , p_4 and p_5 are retrieved by the k-NN query, in this respective order.

In this thesis, the focus is directed to the spatial selection *range*. Given the query location $q.l$ and the distance $dist(p, q.l)$ (euclidean distance between $q.l$ and an object p), the *range* query retrieves 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. The sentence “*find all restaurant in a 100m radius from my actual location*” is an example of the range query. In this spatial query, the objects must satisfy the radius (predicate). Therefore, only objects inside the area defined by the radius are returned by the range query. Figure 2.7(b) illustrates a range query where

$q.l$ is the query location and r is the interest radius. Processing this query on the search space illustrated in Figure 2.7(b), the query returns the points p_3 and p_4 as result.

2.5.2 Spatial Indexes

A spatial database system requires a mechanism to improve the spatial objects retrieval, taking into 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., 2003; SAMET, 1984). Some of these spatial indexes are employed in this thesis and are presented in this subsection.

The R-tree is a balanced tree, almost identical to a B-tree (BARUFFOLO, 1999; BAYER; MCCREIGHT, 2002) 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 reorganizing 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.

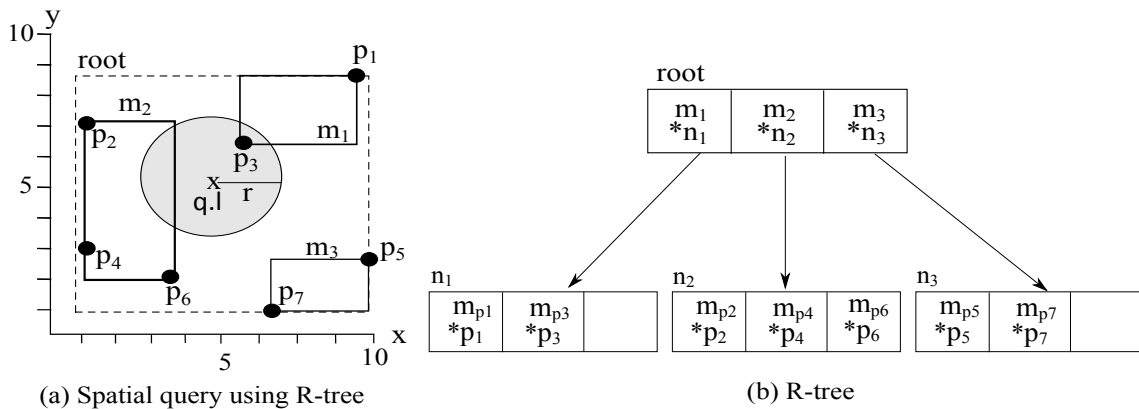


Figure 2.8: R-tree examples. Source: Adapted from Rocha-Junior (2012).

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 an intermediate node that has three intermediate entries m_1, m_2, m_3 pointing 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: $(m_{p_1}, *p_1)$ and $(m_{p_3}, *p_3)$, where m_{p_1} is the MBR involving the spatial object p_1 and $*p_1$ is the pointer (identifier) to object p_1 in the database.

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 is 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 r size. 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 MBR⁸ is inside the spatial neighborhood defined by r , returning object p_3 as consequence.

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 R-tree variations 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). It 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 R-tree variation is the aggregate R-tree (aR-tree), proposed by (PAPADIAS et al., 2003). 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 non-spatial data (e.g. 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 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, returning only those that have a score greater than 0.7 (ROCHA-JUNIOR, 2012; PAPADIAS et al., 2003).

In order to optimize the search 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 score value on the child nodes is added to their respective intermediate node (ROCHA-JUNIOR, 2012; PAPADIAS et al., 2003).

Figure 2.9 represents a Max aR-tree where the aggregated function $max()$ was applied. Therefore, it is possible to observe that the score stored at the intermediate input

⁸In this case, the leaf entries are bi-dimensional points. Thereby, the upper right vertex is identical to the lower left vertex.

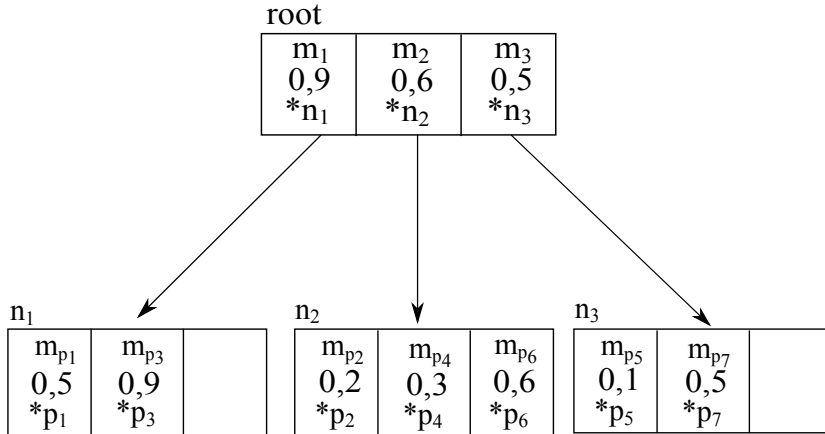


Figure 2.9: AR-tree example.

$(m_1, 0, 9, *n_1)$ is 0.9 because this is the highest score value between the entries of the node n_1 : $(m_{p_1}, 0, 5, *p_1) \in (m_{p_3}, 0, 9, *p_3)$. The structure and the way the aR-tree executes a query 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 in the spatial neighborhood of $q.l$ that 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 .

2.6 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 culminates 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). These preferences are an 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 that 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 is returned by the query. In contrast, if the user requests the list of hotels with a rate higher than 60, the query returns the complete dataset. In this way, the user will have to go through the database until he/she finds the hotel he/she wants, making it difficult to find the cheapest hotel.

Preference Queries (CHOMICKI, 2018) allow the user to express their preferences

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 from 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

more clearly and accurately⁹. One can solve the problem described above by setting the query as follows: “select hotels with the lowest price values, stop after k ” (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).

For example, consider the database described in 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 $f_1(a, b) = [(a[\text{price}] \leq b[\text{price}]) \wedge (a[\text{distance}] \leq b[\text{distance}])]$. 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 $f_1(h_2, h_3) = \text{true}$. All not dominated objects are valid responses to the query described by the formula $f_1(a, b)$.

In contrast, 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

⁹Borzsony, Kossmann and Stocker (2001) demonstrate how to implement a preference query using SQL (without making modifications in the database system). They discuss the reasons for the SQL implementation’s poor performance when compared to an implementation of the preference query that extends the database system with a new logical operator to represent the preference query.

$f(h) = 0.5 * h[\text{rate}] + 0,5 * h[\text{distance}]$, objects with lower scores are those close 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 $h = h[1], \dots, h[n]$, 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 $h_2[2] > h_1[2]$ ($h_2[\text{distance}] = 100$, and $h_1[\text{distance}] = 50$), this function is not monotonic.

2.7 SPATIAL PREFERENCE QUERIES

Spatial databases manage large-size collections of geographic entities. Each entity has geographic coordinates that 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. (2007) present a new type of top- k query - the Spatial Preference Query. In this query, the k best POIs to the user are defined by the quality of the features¹⁰ in the spatial neighborhood of each POI.

Thereby, given a set of POIs P (e.g. candidate locations), the Spatial Preference Query returns the k objects in P with the highest scores. The score of a POI 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 Booking¹¹ where users evaluate various types of features (YIU et al., 2007). The neighborhood is defined by the nearest neighbor, or range, spatial selections.

For example, the white points p in Figure 2.10 represent POIs. In addition, the gray dots represent restaurants while the black dots represent cafeterias. Each restaurant and cafeteria 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. Assuming a tourist wants to get the best hotels in terms of cafeterias and restaurants, the Spatial Preference Query returns the POIs (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 POIs

¹⁰Consider "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.

¹¹<www.booking.com>

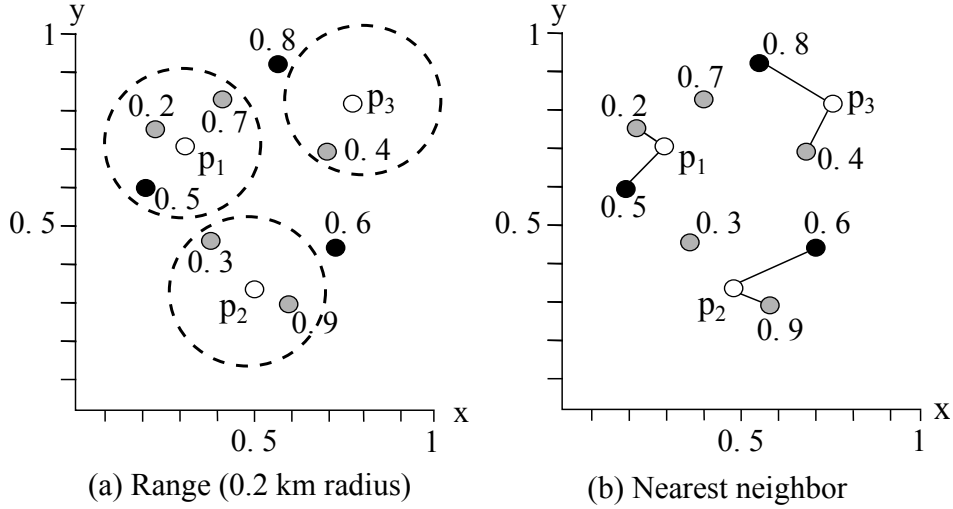


Figure 2.10: Spatial Preference queries examples using different ways to define the spatial neighborhood of a point of interest. Source: Yiu et al. (2007).

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 $\tau(p_3) = 0.4 + 0 = 0.4$. POIs 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, the object p_1 is the top-1 result of the *range* Spatial Preference query. 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 three steps to select the POIs. First, it calculates the distance of the POI to a given feature. Then, it select the features that satisfy the spatial neighborhood definition (range or nearest neighbor). In the end, it orders the POIs by an aggregation function on their scores (YIU et al., 2007). Besides this Top-k query, several works have been 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.

2.8 TOP-K SPATIAL KEYWORD QUERY

Among spatial queries, some 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 index spatial and textual data. These spatio-textual indexes aim to support efficient processing of queries that 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 query response is ordered considering the score values generated for each object by the scoring function.

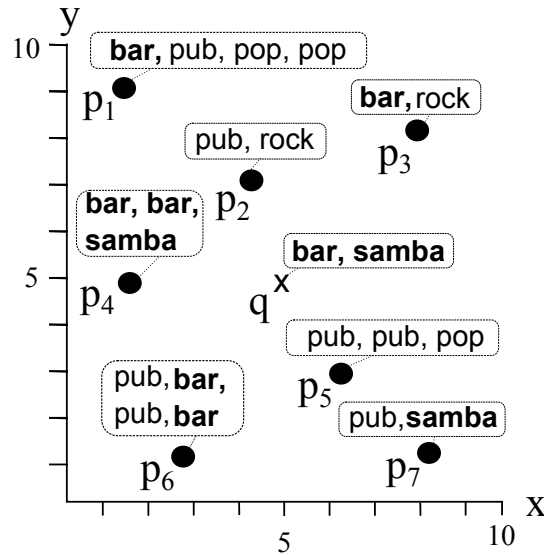


Figure 2.11: Spatial area containing bars and pubs. Source: Rocha-Junior et al. (2010).

Suppose a user wants to find a bar where they have samba presentation in the spatial area described in 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 illustrated in Figure 2.11. In this example, the top- k Spatial Keyword query returns an 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 .

2.8.1 Spatio-textual Indexes

Many applications now use a large amount of spatial data, such as Twitter¹² and Flickr¹³. 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., 2010). 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).

Rocha-Junior et al. (2010) propose a 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., 2010), but it stores the most frequent terms of the collection with a

¹²www.twitter.com

¹³www.flickr.com

different method. S2I maps the most frequent terms of the collection to aggregate R-trees (aR-tree) (PAPADIAS et al., 2003), where each tree stores only objects that have the same term t . Likewise, 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 2.12: Spatial Inverted Index.

The S2I (exemplified in Figure 2.12) 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).

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 $f_{p,t}$ in 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 Minimum 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., 2003). Thus, objects can be accessed decreasingly by $f_{p,t}$ values, and spatial proximity (ROCHA-JUNIOR et al., 2010).

According to Rocha-Junior et al. (2010), 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. 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.

2.9 TOP-K SPATIAL KEYWORD PREFERENCE QUERY

The top-k Spatial Keyword Preference Query (SKPQ) is a query proposed by Almeida and Rocha-Junior (2016) similar to the traditional top-k Spatial Keyword Query (SKQ). A significant part of the traditional spatial queries like the SKQ are user-centered (as discussed in Section 2.8). In other words, they search for spatial objects considering 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.

Different from the user-centered spatial queries, the SKPQ searches for POIs considering other spatio-textual objects (features) in their spatial neighborhood. Specifically, given a set of POIs (e.g. hotels), features set (e.g. bars, restaurants, and tourist attractions), 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 POIs. The score of each POI is given by the highest textual relevance between the query keywords and the text describing the features that satisfy the spatial selection criteria.

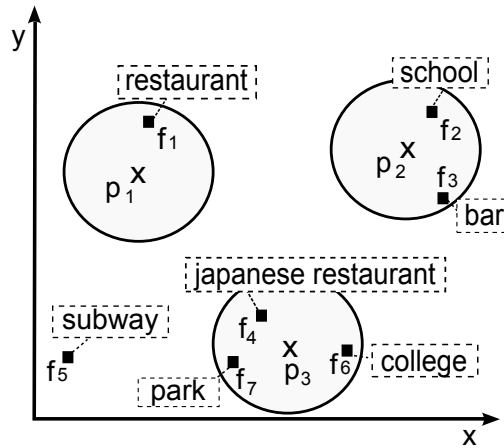


Figure 2.13: POIs (p) and features (f) associated with their textual descriptions.

In essence, 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 2.13 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 textually 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.

2.10 SUMMARY

The queries employed in this research manipulate spatio-textual objects. In other words, the objects are described using a textual description and spatial coordinates (e.g. latitude and longitude). This chapter has presented the underlying concepts of this thesis, as the spatial object and POI definition, together with textual and spatial queries. In addition, the SKPQ is introduced, describing in detail the main query used to manipulate POIs in this research. The review of Spatial Information Retrieval Systems provides background

to address the research question RQ 1 and also contributes to RQ 3. The next chapter addresses the Semantic Web related concepts and describes some of the SPARQL queries employed in this thesis.

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 page content. However, they have no reliable way to process the semantics. The Semantic Web rises as a solution to information systems retrieve the semantics of online content efficiently. It brings structure to the meaningful content of web pages, enabling web applications to answer sophisticated user queries without using 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).

Along with these new data models, it arises the need for a new query language to extract the information. Since the RDF release, several query languages have been proposed (see Haase et al. (2004) for further description). In 2004, the RDF Data Access Working Group released the first draft of SPARQL - a query language for RDF. In essence, Simple Protocol and RDF Query Language (SPARQL) is a graph-matching query language where the query consists of a pattern that 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 advantages of SPARQL are its expressivity and its scalability for large RDF stores thanks to highly optimized SPARQL engines (e.g. Virtuoso, Jena) (FERRÉ, 2014). Query expressiveness determines the type of queries a user is able to pose and how complex is the evaluation of this query. In fact, SPARQL has an expressive power equivalent to Relational Algebra (ANGLES; GUTIERREZ, 2008). In this thesis, SPARQL is used to access Linked Open Data (LOD) repositories to enhance features’ description. An analysis on SPARQL advantages in this scenario is conducted in Section 6.1.6.

The goal of this chapter is to present concepts related to the Semantic Web. It consists of three main sections: i) Section 3.1 presents the basics of RDF, ii) Section 3.2 and 3.3 introduces Ontologies and Linked Open Data, Section 3.4 presents the SPARQL query and illustrates how to 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 that enable deductions in RDF data. RDF represents information in a minimalist and flexible way. RDF usually shares information between applications that have distinct design setups. This framework increases the value of information as it becomes accessible to more applications across the entire Internet (KLYNE; CARROLL, 2006).

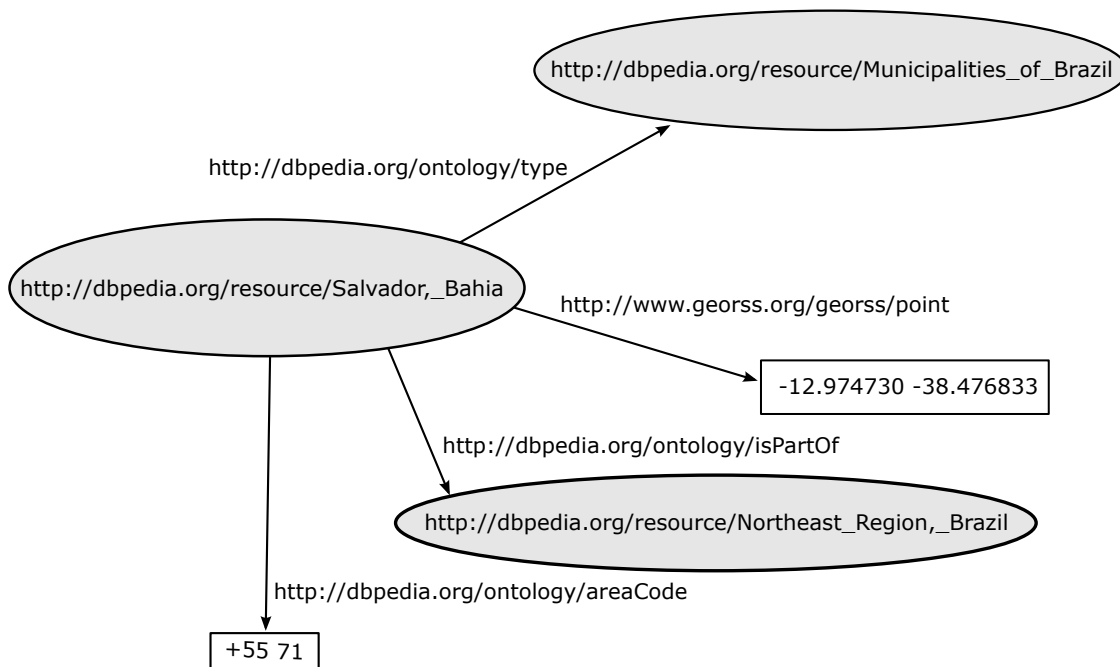


Figure 3.1: An example of an RDF graph describing the city of Salvador.

The RDF structure is a collection of triples containing 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) (PAN, 2009).

The Uniform Resource Identifier (URI) is employed to identify the resources described in RDF. When an RDF node has a URI label on it (like the gray ones in Figure 3.1), the URI identifies the resource represented by the node. Consequently, RDF assumes that nodes with the same URI represent the same resource (KLYNE; CARROLL, 2006).

Each triple expresses a statement of a relationship between nodes. Each triple has three parts: a subject, an object, and a predicate (also known as property) that denotes a relationship. The nodes of an RDF graph represent the subjects and objects, while the

arcs are the predicates. The arc always points toward the object. For instance, Figure 3.1 exemplifies an RDF graph describing a city identified by:

- the subject `<http://dbpedia.org/resource/Salvador,_Bahia>`;
- the subject type `<http://dbpedia.org/resource/Municipalities_of_Brazil>`;
- its spatial coordinates `"-12.974730, -38.476833"`;
- the region it is located: *Northeast of Brazil* (represented by `<http://dbpedia.org/resource/Northeast_Region,_Brazil>`);
- and its respective area code `+55 71`.

The predicates are represented by the URIs near the arcs (e.g. `<http://www.georss.org/georss/point>`), and the objects are represented by values inside the rectangles or ellipses. An object can be a literal (e.g. `+55 71`), an RDF URI reference (`<http://dbpedia.org/resource/Municipalities_of_Brazil>`) 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 and Carroll (2006), the assertion of an RDF graph amounts to asserting all the triples in it. Therefore, the RDF graph meaning is the conjunction (logical AND) of the statements corresponding to all the triples it contains.

3.2 ONTOLOGIES

An 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 structure concept to build shareable and reusable LOD repositories. 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 concept relationships.

Ontology classifies things in terms of semantics or meaning. In OWL, this is achieved by using 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 *Mouse* as subclasses of *Animal*, and *Tree* and *Grass* as subclasses of *Plant*. The individuals are members of a given OWL class, thereby we

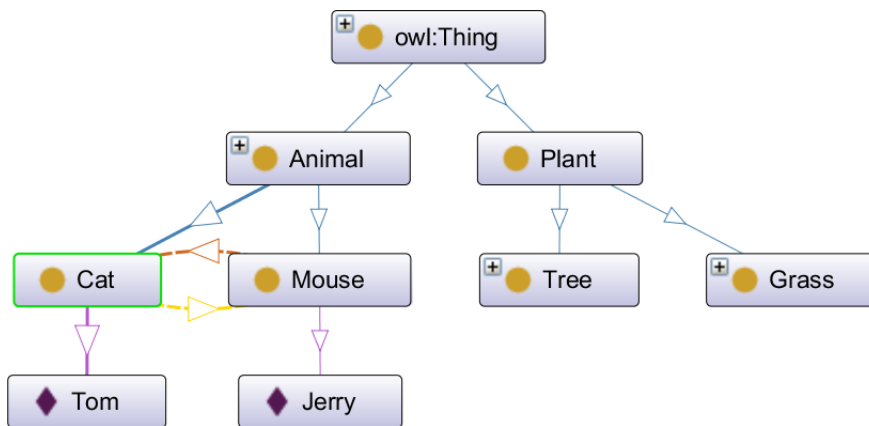


Figure 3.2: An ontology representing concepts and relationships between concepts.

can define “Tom” as a member of the *Cat* class. This way, we can infer that “Tom” is an animal too because *Cat* is a subclass of *Animal* in the ontology graph.

There are two types of property in OWL to which an individual is related: i) object properties (i.e. `owl:ObjectProperty`) relate individuals of two OWL classes, and ii) datatype properties (i.e. `owl:DatatypeProperty`) relate individuals (instances) of OWL classes to literal values. For instance, it is possible to create an object property to describe that *Cat* eats *Mouse* as described in Listing 3.1 and 3.2. First, it is defined the relationship *eats* using `<owl:ObjectProperty>` (Listing 3.1), then `<owl:Class>` defines the class *Cat* while the `<owl:Restriction>` defines that every instance of *Cat* eats an instance of *Mouse* (Listing 3.2). Therefore, it is possible to infer that Tom eats Jerry. The relationship *eats* is represented by the yellow arc, while the relationship *eaten_by* is described by the red arc in Figure 3.2.

```

<!-- http://semantic.org/people#eats -->

<owl:ObjectProperty rdf:about="http://semantic.org/animal#eats">
  <rdfs:domain rdf:resource="http://semantic.org/animal#animal"/>
  <rdfs:comment></rdfs:comment>
  <rdfs:label>eats</rdfs:label>
</owl:ObjectProperty>
  
```

Listing 3.1: Object property representing the relationship “eats”.

```

<!-- http://semantic.org/animal#cat -->

<owl:Class rdf:about="http://semantic.org/animal#Cat">
  <rdfs:subClassOf rdf:resource="http://semantic.org/animal#Animal"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="http://semantic.org/animal#eats"/>
      <owl:allValuesFrom rdf:resource="http://semantic.org/animal#
        Mouse"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment></rdfs:comment>
  <rdfs:label>Cat</rdfs:label>
</owl:Class>

```

Listing 3.2: Cat class definition including that Cat eats Mouse.

In the same fashion, it is possible to create a datatype property defining the number of legs to an animal, as described in Listing 3.3. First, it is set the property using `<owl:DatatypeProperty>`; then the property is used to relate the individual *Jerry* with the literal value 4 representing his number of legs.

```

<!-- http://semantic.org/animal#legs -->

<owl:DatatypeProperty rdf:about="http://semantic.org/animal#legs">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</owl:DatatypeProperty>

<!-- http://semantic.org/animal#Jerry -->

<owl:NamedIndividual rdf:about="http://semantic.org/animal#Jerry">
  <rdf:type rdf:resource="http://semantic.org/animal#Mouse"/>
  <legs rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">4</legs>
</owl:NamedIndividual>

```

Listing 3.3: Datatype property describing the number of legs in an animal.

It is important to realize that RDF defines the data structure, while OWL describes semantic relationships. RDF allows the user to link concepts together; therefore, it is possible to describe that the Salvador’s (concept) area code is +55 71 (another concept), as Figure 3.1 describes. In brief, the triples describe a single fact: “Salvador, _ Bahia areaCode +55 71”. However, it is not possible to classify objects using RDF. For example, it is not possible to infer that “Municipalities _ of _ Brazil” is a subclass of the Brazil class.

OWL is a more expressive knowledge representation than RDF. It categorizes properties (relationships) into object and datatype properties, enabling the user to add re-

restrictions on properties. For example, it is possible to define *Tom* as a *Cat*, and infer that *Tom* eats *Jerry* because every *Cat* eats every *Mouse* in the Class definition. This information is not possible to obtain using RDF.

Actually, OWL is in its second version (OWL 2) that extends the OWL 1 to facilitate ontology development and sharing. OWL 2 has a similar overall structure to OWL 1, but it adds new functionality like new constructs for properties, extended support for datatypes, and extended annotations.

3.3 LINKED OPEN DATA - LOD

The Web has evolved into a space where both documents and data are linked (BIZER; HEATH; BERNERS-LEE, 2011). The Semantic Web is an extension of the World Wide Web that aims to make the information available online machine-readable. In order to support this new Web, a set of practices for publishing and connecting structured data has been proposed by Berners-Lee (2006). 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 (2006) 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 (e.g. RDF, SPARQL);
4. Include links to other URIs, enabling users to 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 that points into another file. Suppose an RDF file, named `<http://example.org/Hotels>`, that describes hotels around the world. Listing 3.4 exemplifies the RDF description of the hotel Danieli. Local identifiers (Venice, Italy, and Hotel_Danieli) are used to describe the hotel (resource). 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 3.4: Description of hotel Danieli in an RDF file.

Now, suppose there is another RDF file (listing 3.5) describing Hotels in Venice. Hotel Danieli is in Venice; however, there is no need to define it again in Listing 3.5. Hotel Danieli is described by its HTTP URI that points to its description. When these files are released under an open license, they are called Linked Open Data (LOD). In this thesis, we use two LOD sources: DBpedia and LinkedGeoData. The use of these sources is detailed by Chapter 6, in the dataset section of each framework’s module, which describes how these repositories were accessed and used to process the query. Furthermore, the DBpedia is introduced in the next section.

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

Listing 3.5: Description of hotels in Venice in a RDF file.

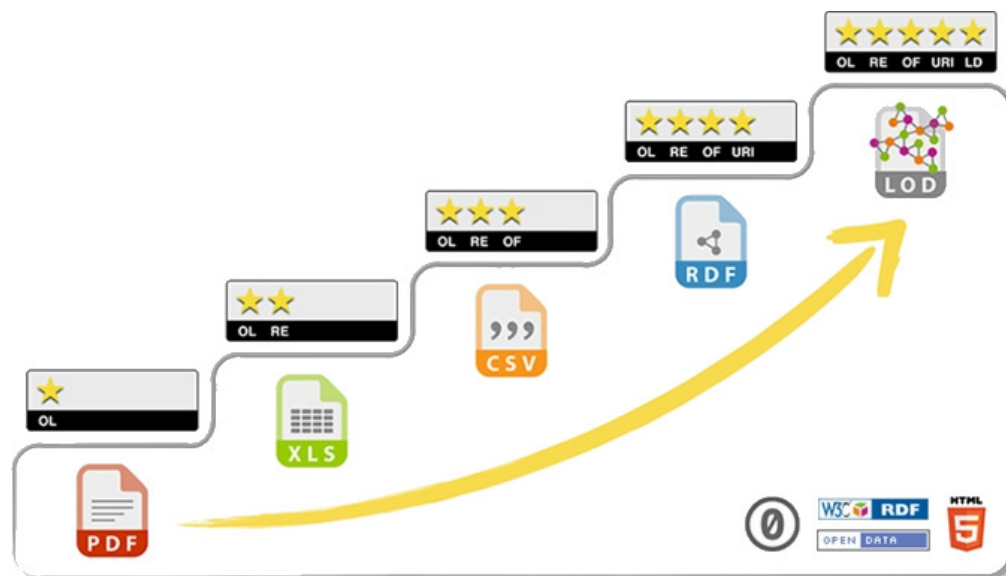


Figure 3.3: Five Star Scheme for LOD. Source: Kim and Hausenblas (2019).

Berners-Lee (2006) suggested a 5-star rating system for Linked Open Data, illustrated in Figure 3.3. 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, an Excel file instead of an image scan of a 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 repositories. In 2019, this project connected 1,239 repositories with 16,147 links between them (MCCRAE, 2019), resulting in more than 31 billions items (FRESSATO, 2019). This collection of repositories is known as the LOD cloud. As a result, web search engines can use HTTP URIs to access data within different LOD repositories, 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).

In the first quarter of 2020, it is possible to access a significant number of LOD repositories representing different natures of the data (MCCRAE, 2019). For example, the GeoNames and the LinkedGeoData make it possible to add geospatial semantic information to the World Wide Web. Several other sources provide government data as an RDF knowledge base, such as the Eurostat Linked Data and the World Bank Linked Data. DBpedia is another important LOD source because it allows sophisticated queries against Wikipedia and links datasets on the Web to Wikipedia data. In this D.Sc. thesis, the DBpedia and LinkedGeoData are used to process spatial queries aiming to improve query-user satisfaction.

3.3.1 DBpedia

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 repository, one must submit a query using SPARQL language. For this reason, the endpoint usually is called the SPARQL endpoint. One can ask queries against DBpedia using the OpenLink Interactive SPARQL Query Builder (iSPARQL)¹, the SNORQL query explorer², or any other SPARQL-aware client(s). In this research, we use ARQ³ to access DBpedia. ARQ is a SPARQL processor for Jena - a free open source framework for building Semantic Web and Linked Data applications.

The DBpedia latest data core (2020) is the refurbished equivalent of the previous releases (e.g. 2016). The most recent statistics report (October 2016) states that the English version of the DBpedia knowledge base describes 4.9 million things (instances with abstracts), including 1.5 million persons, 840,000 places, 496,000 creative works (music albums, films, and video games), 286,000 organizations (companies and educational institutions), 306,000 species, 58,000 plants, and 6,000 diseases. In addition, DBpedia

¹<<http://dbpedia.org/isparql>>

²<<http://dbpedia.org/snorql>>

³<<https://jena.apache.org/documentation/query/service.html>>

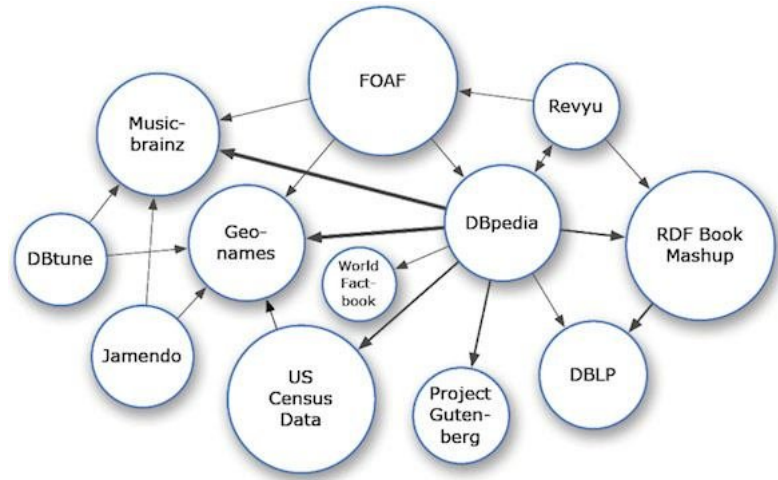


Figure 3.4: Example of the cross-domain LOD subcloud with DBpedia and related datasets. The updated LOD subcloud is available at: <https://lod-cloud.net/clouds/cross-domain-lod.svg>.

provides localized versions in 125 languages. All these versions together describe 13 billion things (FREUDENBERG, 2019). The DBpedia data core (version 2016-10) was updated in January (2021).

DBpedia has advantages over existing knowledge bases like wide domain coverage, the information is generated from real community agreement, automatically evolves as Wikipedia changes, and it is truly multilingual. For these reasons, the DBpedia is employed in this thesis to enhance the textual description of POIs. The enhancement algorithm is described in detail by Chapter 5. Figure 3.4 exemplifies the cross-domain LOD subcloud highlighting DBpedia and related datasets (MCCRAE, 2019). A full LOD image can be accessed in <https://lod-cloud.net/clouds/cross-domain-lod.svg>.

3.4 SPARQL

SPARQL is a query language that can 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 enables 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>. This research employs SPARQL to search for POIs and to enhance their textual descriptions.

SPARQL contains capabilities for querying graph patterns along with their conjunctions and disjunctions. Essentially, a SPARQL query consists of a pattern that is matched against a data source, and the values obtained from this matching are processed to give the answer. The SPARQL query result can be a result set or an RDF graph. Listing 3.6 introduces a SPARQL query to obtain objects within a 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 3.6, 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 20 km as illustrated at Listing 3.6.

```
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 3.6: SPARQL query to obtain objects within 20 km radius of New York city.

3.5 SUMMARY

This chapter presented an overview of the Semantic Web and described how SPARQL is used to process a spatial query. First, it started by introducing the Resource Description Framework. Then, Linked Open Data is discussed as one of the core concepts of this new Web. Finally, SPARQL is presented, describing with listings how to use it to find spatial objects in a search space. This chapter presented the underlying concepts to address the research question RQ 2. The next chapter presents a literature review on textual enhancement using LOD, query personalization, and ranking functions.

PART III

**EXPLOITING OPEN DATA FOR
IMPROVING SPATIAL KEYWORD
QUERY APPLICATIONS**

RELATED WORK

Each module of the framework implements a technique to improve the order of items in the query result. In this chapter, we review and compare related works that propose techniques similar to each proposed module. This literature review guides our answer to the research questions: How can we order the best POIs retrieved by spatial preference queries to satisfy the user (RQ 1)?, How to exploit LOD to process spatial preference queries (RQ 2)?, and How to model the user preference to improve SKPQ results (RQ 3)?. Also, this chapter describes techniques and strategies to achieve the Specific Objectives described in Section 1.3.1. Section 4.1 describes the literature review about the Description Enhancement using LOD, Section 4.2 discusses the studies about spatial query personalization, and Section 4.3 report the works employing probabilistic functions to model the user preference.

4.1 DESCRIPTION ENHANCEMENT USING LOD

Figure 4.1 presents the related works described in this section following a chronological order. Each work describes a strategy to enhance the POI's description using Linked Open Data (LOD). In Figure 4.1, each milestone corresponds to the article's name (in short or adapted), its year of publication, and its citation in this thesis. Our proposal is highlighted by the yellow star symbol.

Studies employ LOD datasets to improve textual descriptions of spatial objects (FERNÁNDEZ-TOBIÁS et al., 2011; HEGDE et al., 2011; KARAM; MELCHIORI, 2013). Hegde et al. (2011) describe an augmented reality browser that uses LOD to enhance the description of POIs. These objects are represented by a semantic relationship between them and several spatial data repositories such as Wikipedia and YouTube. Using Natural Language Processing techniques, the user's profile is semantically related to a set of POIs. Then, the personalized set of POIs is delivered to the user. Similarly, (KARAM; 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 POIs from different LOD sources. In M-PREGeD, voluntary users can generate or update POIs'

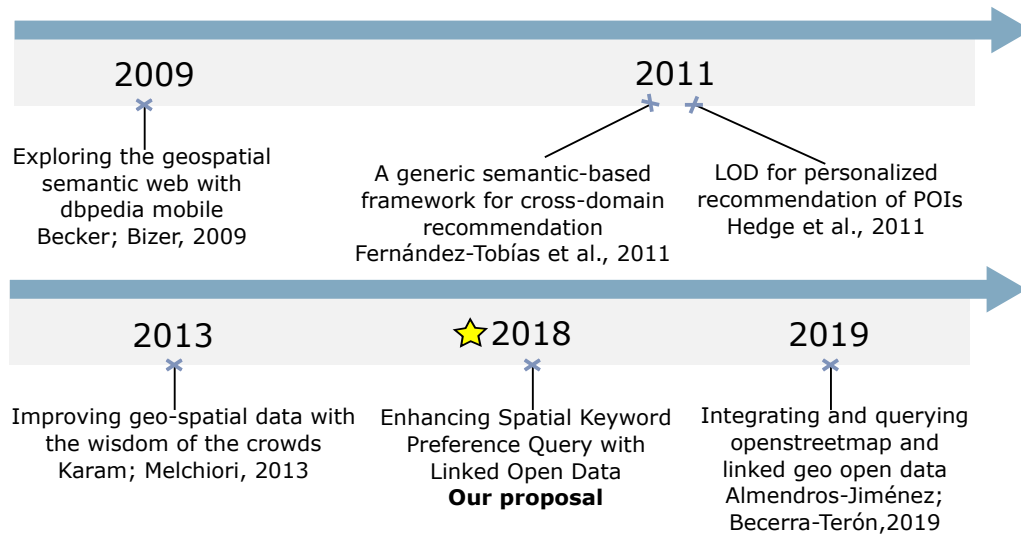


Figure 4.1: Timeline of related works on description enhancement of POIs.

descriptions to enhance them. Aiming the same goal, we use DBpedia to enhance the textual description of features. However, we do not use 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 to use 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 location. Likewise our approach, they used LOD to obtain data about a spatial area (e.g. architecture in Rome), but they did not make use of any spatial information (e.g. latitude or longitude) in their recommendation. We use spatial information (coordinates of a feature) to select objects that satisfy the user’s information need.

Equally important, Becker and 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 information is obtained by navigating along with data links into different data repositories. In this thesis, we use the semantic representation of spatial objects available at DBpedia to measure the similarities between the user’s keywords and the feature. Thereby, we automatically combine information from different LOD repositories instead of enabling the user to navigate freely through the repositories. Moreover, LOD is employed during query processing in this research. In this stage, the user interaction is expected to be minimum.

Accordingly Braun, Scherp and 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. A revision engine based on data mining techniques is provided to improve the POI quality. 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 LOD as well as create and collaboratively edit thematic maps. Despite their tool does not provide a revision engine to improve POI quality; it provides an exploration of LOD that span across multiple SPARQL endpoints. By querying these endpoints, the user can create his maps and share these maps with others. The LOD is explored by a tool that builds a class hierarchy and discovers the spatial extent of available information. This thesis uses 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 in a 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 thesis, we aim to enrich the data that describes a spatial object. In a like manner, query expansion has long been suggested for dealing with the word mismatch issue in information retrieval. Accordingly, to Xu and Croft (2017), there is a number of query expansion approaches. The main approaches are to analyze the query description to discover word relationships (global techniques) and to analyze the objects retrieved by the query location (local feedback) (XU; CROFT, 2017).

With this in mind, Karpathiotaki et al. (2014) introduce the Prod-Trees platform, a semantically enabled search engine for earth observation products (e.g. 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 he/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 this thesis, the user can submit free-text queries as well as in the Prod-Trees platform, but we do not use a query analyzer to expand the query. In contrast to query expansion, this thesis investigates strategies to improve the description of spatial objects.

Data integration is also applied to improve the description of POIs. Almendros-Jiménez, Becerra-Terón and Torres (2019) propose a framework to convert spatial data in RDF (e.g. DBpedia data) into OpenStreetMap (OSM) format. It can access a LOD dataset and transform its data into the OSM format. Then, the framework can integrate both the OSM data and the transformed data. This way, it is possible to add information into the OSM dataset, providing a description for POIs that does not have it yet. In contrast, we concatenate the existing OSM description with the one accessed in DBpedia instead of adding new information about the POI or adding new POIs. Moreover, our proposal does not have to transform the LOD data into OSM to enhance the textual description. All enhancement process is done through SPARQL queries automatically.

Table 4.1 lists the features of several approaches, including our proposal to enhance the textual description of features. It is important to notice that the novelty of this work is to employ already existing enrichment text techniques based on LOD to improve SKPQ processing. To the best of our knowledge, there is no similar improvement applied to a top-k Spatial Keyword Preference query.

Table 4.1: Characteristics employed by different approaches to enrich textual descriptions. Linked Open Data (LOD), Spatial Objects (SO), Voluntary Users (VU), Natural Language Processing (NLP), Metadata (MD), and Data Integration (DI).

Methods	LOD	SO	VU	NLP	MD	DI
Proposed approach	✓	✓				
Becker and Bizer (2009)	✓	✓				
Hegde et al. (2011)	✓	✓		✓		
Fernández-Tobías et al. (2011)	✓	✓				
Karam and Melchiori (2013)	✓	✓	✓			
Almendros-Jiménez, Becerra-Terón and Torres (2019)	✓	✓				✓

4.2 PERSONALIZATION OF SPATIAL QUERIES

Location-Based Services (LBS) are aggregating relevant information about users, their behavior, and their preferences based on the location histories (GASPARETTI, 2017). By personalizing these services, users can obtain information that matches their preferences and goals. This section presents an overview of the previous work on personalization methods in LBS. Figure 4.2 illustrates in chronological order the related work described in this section. Each milestone corresponds to the article’s name (in short or adapted), its year of publication, and its citation in this thesis. The article describing our personalization approach is highlighted by the star symbol.

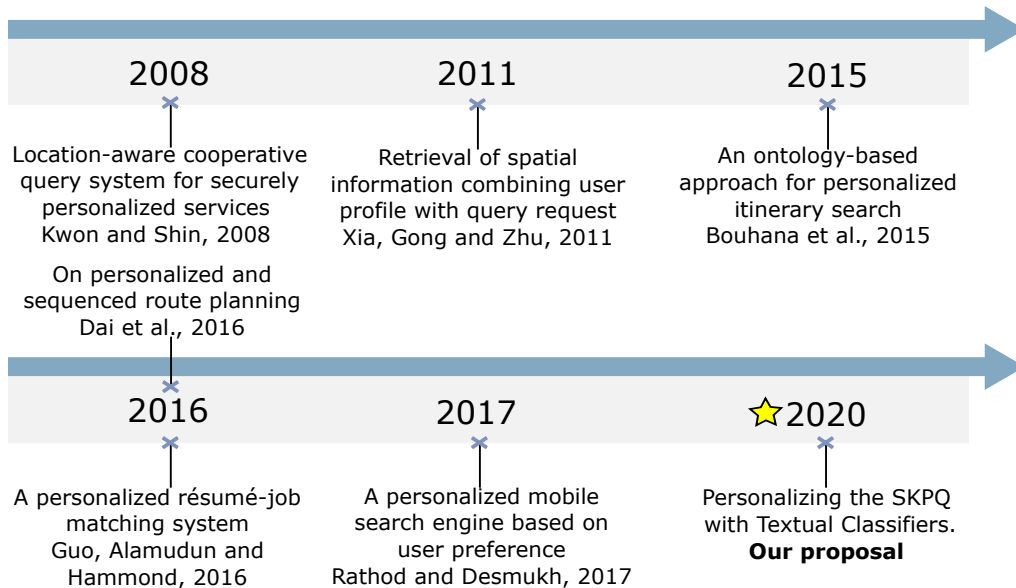


Figure 4.2: Timeline of related works on personalization of spatial queries.

Some approaches have been proposed to deal with the personalization of query results. Kwon and Shin (2008) propose a personalized location-aware query methodology based

on the current location and schedule of the user. During the search process, they apply a contextual concept distance to assess when the user is interested in visiting a POI. The user's schedule is used to personalize the query results, presenting only POIs that are relevant to his/her scheduled activities. Likewise, our approach personalizes the query results to increase the rank position of points similar to those the user enjoyed in the past. However, we employ the user's reviews instead of their schedule to achieve the same goal.

Moreover, Kwon and Shin (2008) process a location-aware cooperative query that does not have to satisfy a k constraint defining the number of POIs which must be returned to the user. The possibility to consider the user preferences to reduce (or increase) the number of POIs enables the system not only to select the best POI for the user but to define the best number of results too. An ideal k value includes all relevant objects to the user in the query result. Different from Kwon and Shin (2008), our approach also has the challenge to satisfy the k value defined by the query user.

Several studies dynamically enhance queries building user profiles to provide answers more accurate to the user preferences. Xia, Gong and Zhu (2011) employ user profiles to personalize spatial information retrieval. The user profile is composed of query conditions and refined information obtained from the dataset. In this personalization model, the system builds a user profile every time a user submits a query. Conversely, we build a user profile before the query processing, and this profile contains only information given by the user in his past interactions with the system.

Margaris, Vassilakis and Georgiadis (2018) apply user profiles to personalize movie search results. They employ collaborative filtering techniques and take into account the influence factors between social network users. An offline phase initializes the user profiles with information about the user (i.e. movies ratings) and user relationships (friends who have an influence mathematically described on the user). Then, the query is processed taking into consideration the user's friends decisions to personalize the query results. In our scenario, we do not know who is the friend of the query user. Therefore, we explore similarities between the user's reviews and the reviews describing the POI to identify the POI that best satisfies the query user.

Boudighaghen, Tamine and Boughanem (2011) personalize Web search results considering the user's location. First, they identify which query is location-aware using classifiers like Decision trees, Naive Bayes, and Support Vector Machines. Then, they explore two alternatives to personalize the query: query expansion and query results re-ordering. The query expansion proposes including the user location as a query keyword. It processes the expanded query instead of the one submitted by the user. In sequence, they re-order the query rank, increasing the ranking position of results that contain at least one word describing the user location (i.e. city name). Different from Boudighaghen, Tamine and Boughanem (2011), our query searches for points of interest (POIs) instead of Web pages. POIs have spatial coordinates that enable our system to accurately identify the location of the result. In this way, instead of applying classifiers to identify the type of query, we use classifiers to learn the user preferences.

Rathod and Desmukh (2017) propose a Personalized Mobile Search Engine (PMSE) that employs the user's past rating and clickthrough data. In this system, the query results are presented to the user without any personalization. The user is asked to rate

results that he/she judges relevant manually. Then, the clicks on the results and the judgments provided by the user are used to personalize and re-order the query results. This approach demands considerable user inputs to personalize the query results. According to Kwon and Shin (2008) and Allan et al. (2012), a location-aware service should ask for the minimum user input as possible. Likewise, our approach uses just the query keywords to describe the user need and the user reviews to personalize the query result. Additionally, our personalization method does not ask the user to submit any information manually during the query processing. While they use only an explicit interaction (user ratings) to personalize their system, our solution uses only implicit (user reviews) interactions to achieve the same goal.

Guo, Alamudun and Hammond (2016) developed the Résumatcher - a personalized system to find jobs. The user submits his/her résumé in the system; then it presents jobs descriptions that contain skills similar to those described in his/her 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 a complex task that requires a large and robust information system. Aiming at this problem, Bouhana et al. (2015) present an information retrieval method to personalize itinerary searches in urban freight transport systems. In this system, the user submits a request containing topics to describe his/her preferences. Then, they combine a Case Base Reasoning (CBR) and the Semantic Web Rules Language (SWRL) 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. Our approach uses just the query keywords to describe the user need and user reviews to personalize the query result.

User profiling is widely used in Recommender Systems, Image Retrieval Systems, and Web Search to personalize the interaction with the user (CHIVADSHETTI; SADAFALÉ; THAKARE, 2015; GASPARETTI, 2017; ZEMEDE; GAO, 2017). However, the application of this tool to personalize spatial queries still is relatively sparse. To the best of our knowledge, there is no other personalization model that employs textual classifiers to learn the user preferences aiming to re-order query results from spatial queries. Table 4.2 presents a comparison based on features of other location-aware information systems. Several information systems employ a personalization model in the search process. However, few systems associate personalization with top-k queries. In fact, no one personalizes top-k queries considering only implicit user interaction.

4.3 PROBABILISTIC FUNCTIONS IN QUERY PROCESSING

Probabilistic functions have been used in different fields. In query processing, the probabilistic functions have been used for estimating the query results when the data is un-

Table 4.2: Features comparison between our approach (P-SKPQ) and other Spatial Information Retrieval systems.

Spatial Information Retrieval Systems	Personalization model	Top-k query	User Interaction
SKPQ	No	Yes	No
P-SKPQ	Yes	Yes	Implicit
Kwon and Shin (2008)	Yes	No	Implicit
Xia, Gong and Zhu (2011)	Yes	No	Implicit
Bouhana et al. (2015)	Yes	No	Explicit
Guo, Alamudun and Hammond (2016)	Yes	No	Explicit
Dai et al. (2016)	Yes	No	Explicit
Rathod and Desmukh (2017)	Yes	Yes	Explicit

certain (CHENG; KALASHNIKOV; PRABHAKAR, 2003), modeling the user mobility (ZHANG et al., 2016), applying the learning to rank method for optimizing search engine results (KUZI et al., 2019), and associating documents’ keywords to knowledge graphs topics (MENG; LI; ZHANG, 2020). In this thesis, we employ the probabilistic function to describe the user’s spatial preference in the ranking function.

4.3.1 Rank Based on probabilistic functions

Figure 4.3 depicts the related works described in this section in chronological order. Each milestone corresponds to the article’s name (in short or adapted), its year of publication, and its citation in this thesis. The proposal of our novel ranking function, which considers a probabilistic function, is highlighted by the star symbol.

The Location Promotion Problem refers to ranking users according to the visiting probability to a target POI. One solution to this problem includes modeling user mobility through the user’s check-in behavior (FENG et al., 2017). For example, Feng et al. (2017) adapt the word2vec algorithm to model the users’ check-in sequence and predict the user visiting probability in a POI. Likewise, Zhang et al. (2018) analyze the users’ check-in distribution and employ embedding vectors to integrate different preferences in a unified preference model. Based on the embedding vectors, Zhang et al. (2018) adopts the Pareto distribution to estimate the user visiting probability. Despite adopting a probabilistic function to estimate user preference such as Feng et al. (2017) and Zhang et al. (2018), our approach differs on the problem objective. Our approach retrieves a set of POIs that satisfy a target user instead of a set of users to visit a target POI.

An effective spatial preference model integrates a combined effect of multiple factors, such as user interest, POI popularity, and distance between the user and POI (LIU et al., 2013). Liu et al. (2013) describe a geographical-probabilistic analysis to model these factors. They apply a multinomial and a Gaussian distribution to model the user mobility

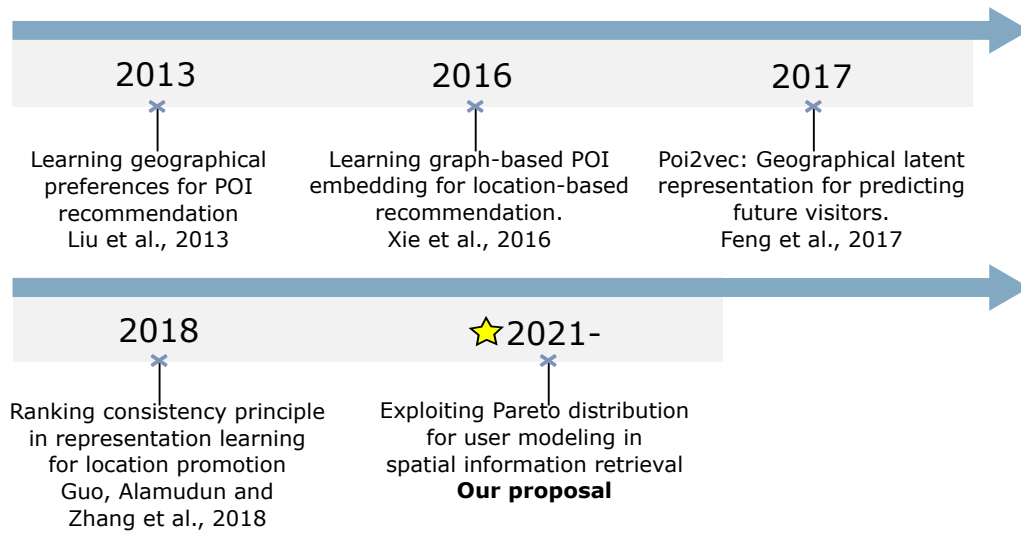


Figure 4.3: Timeline of related works on probabilistic functions to model the user preference.

over clusters of POIs. Xie et al. (2016) propose a generic graph-based embedding model to capture factors, such as the user mobility pattern and spatial preference, the time effect on user behavior, and the POIs' descriptions. Different from Liu et al. (2013), we adopt Pareto instead of the Gaussian distribution because it is hard to evaluate the probability of a user visiting a location based on bivariate Gaussian mobility models (ZHU et al., 2015). Unlike Xie et al. (2016), we do not adopt embedding vectors since they require time to learn the user preference and require updates to reflect the current user preference. Our thesis focus on determining a ranking function to estimate the user preference for an unknown POI using distance distribution.

Table 4.3 presents a comparison between features in Spatial Retrieval Systems that employ probabilistic rank functions, our novel probabilistic rank function, and the SKPQ rank function. The majority of studies adopt probabilistic functions coupled with embedded vectors. However, they do not use $top-k$ queries to retrieve the data. They are recommender systems or user models employed to classify and rank unknown POIs.

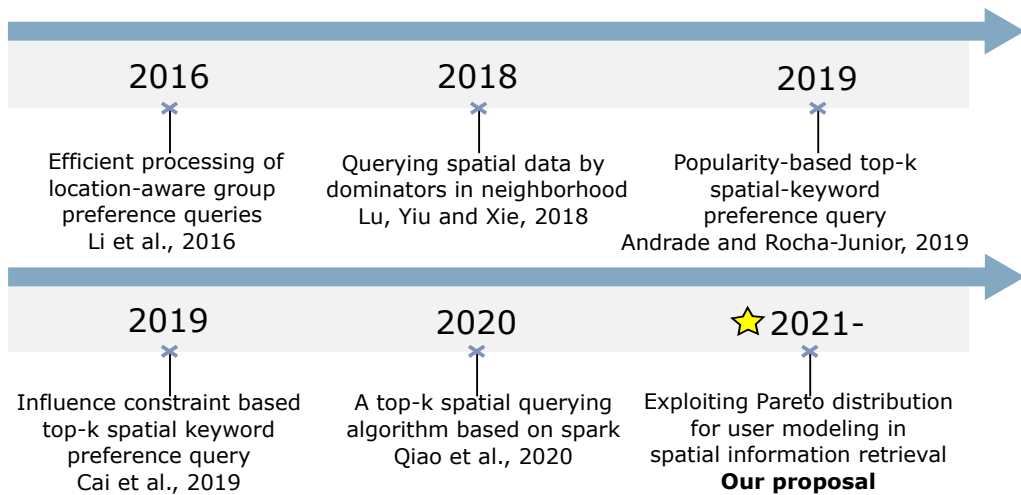
4.3.2 Ranking on $Top-k$ Spatial Preference Queries

Figure 4.4 presents the related studies described in this section following a chronological order. Each milestone corresponds to the article's name (in short or adapted), its year of publication, and its citation in this thesis. The proposal of our probabilistic-based ranking function is highlighted by the star symbol.

The massive amount of spatial data available online brings opportunities to discover suitable locations for the user. Conventionally, spatial queries select objects solely based on their locations and distances. Typical spatial queries include range queries (YIU et al., 2007; ZACHARATOU et al., 2019), nearest neighbor queries (LE et al., 2019; YIU et al., 2007), or spatial joins (LI; TANIAR, 2017; QIAO et al., 2020). $Top-k$ spatial preference

Table 4.3: Comparison between features in Spatial Retrieval Systems that employ probabilistic rank functions and our novel probabilistic rank function.

Spatial Information Retrieval Systems	Top- k query	Probabilistic Function to rank	Embed Vectors
SKPQ	Yes	No	No
Probability-based rank function	Yes	Yes	No
Liu et al. (2013)	No	Yes	Yes
Xie et al. (2016)	No	Yes	Yes
Feng et al. (2017)	No	Yes	Yes
Zhang et al. (2018)	No	Yes	Yes

**Figure 4.4:** Timeline of related works on ranking functions in top- k spatial preference queries.

query is a class of query that ranks POIs based on their spatial neighborhood. Recent proposals on this class of query focus on query processing performance (QIAO et al., 2020; ZACHARATOU et al., 2019) or query variations (ANDRADE; ROCHA-JUNIOR, 2019; LE et al., 2019). Different ranking functions have been proposed in these works: Qiao et al. (2020) apply a ranking function on pairs of POIs considering their distance and score; Andrade and Rocha-Junior (2019) count the number of features in the POI vicinity; Lu, Yiu and Xie (2018) analyze the POI’s spatial location and the quality of their attributes (e.g. POI rating); Li et al. (2016) consider the maximum and the minimum distance between a POI and a group of features; and Cai et al. (2019) employ the influence ranking function that decreases the score value while the distance between the POI and the feature increases. To the best of our knowledge, our proposal is the first to consider the Pareto distribution in the ranking function to describe the user’s spatial preference.

Nowadays, users are more aware of their sensitive information (e.g. user location

or preferences) that are analyzed or revealed by Location-based services. Crimes like harassment, car theft, or kidnapping could occur whether a criminal has access to the user’s sensitive information (ZHU; LIU; LI, 2017). Thereby, solutions to design secure and efficient spatial queries have attracted interest recently (JADALLAH; AGHBARI, 2019; KIM; KIM; CHANG, 2019; ZHANG et al., 2019; ZHU; LIU; LI, 2017). Three techniques are frequently applied in the literature to preserve the user privacy: cloaking technique (JADALLAH; AGHBARI, 2019), k -anonymity (ZHANG et al., 2019), and homomorphic encryption (KIM; KIM; CHANG, 2019). “Cloaking” refers to remove the user identity and cloak (blur) the user location with a circle. k -anonymity means that the query user shares the circle that blurs his/her location with other $k - 1$ users. The homomorphic encryption enables an outsourced server to process encrypted data and return encrypted results. After removing the encryption, the results are the same as they had been processed without encryption. Our algorithms do not apply any of these techniques because they do not require user identity or location, guaranteeing user confidentiality. Table 4.4 describes a comparison between features in spatial queries and our novel probabilistic rank function. Among the top- k queries and keyword queries, we did not find any that employs probabilistic functions to rank POIs.

Table 4.4: Comparison between features in Spatial queries and our novel probabilistic rank function.

Spatial Information Retrieval Systems	Top-k query	Keyword query	Probabilistic function to rank
Probability-based rank function	Yes	Yes	Yes
Li et al. (2016)	Yes	No	No
Lu, Yiu and Xie (2018)	Yes	No	No
Cai et al. (2019)	Yes	Yes	No
Andrade and Rocha-Junior (2019)	Yes	Yes	No
Qiao et al. (2020)	Yes	No	No

4.3.3 Out of scope: Pareto Curve

It is important to disambiguate our proposal from the Pareto curve (CARAMIA; DELL’OLMO, 2020; LIU et al., 2015), also known as Skyline operator (BORZSONY; KOSSMANN; STOCKER, 2001). Pareto curve is frequently applied to process top- k spatial queries efficiently (CHANG; CHEN; CHUANG, 2019; LIU et al., 2015; ROCHA-JUNIOR et al., 2010). Specifically, a solution on the Pareto curve is a candidate solution that is not dominated by any other possible solution. A POI p dominates another POI p' when p is not worse than p' in any attribute and p is better than p' in at least one attribute (e.g. price or rating) (ZHIMING; AREFIN; MORIMOTO, 2012). Applying the dominance concept, it is possible to process the query without access every POI in the dataset. Extensions of this operator have been proposed, such as the G-Skyline (LIU

et al., 2015) that applies the operator in a group of POIs, or enabling the support of efficient top- k spatial preference queries processing (ROCHA-JUNIOR et al., 2010). In this thesis, we do not focus on query processing efficiency. Thereby, we do not propose new indexes or aim to identify a minimal subset of POIs to prune the search space. The probability-based rank function is based on the assumption that users prefer to visit POIs whose user's desired feature is in a short distance. Under this assumption, we model this user preference using the Pareto distribution.

4.4 SUMMARY

This chapter presented a literature review that unveiled related articles concerning POI description enhancement, query personalization, user modeling, spatial queries, and rank functions. As shown, LOD has been used to improve the description of places in different scenarios, considering the user feedback when needed. This review contributes to understanding how to employ LOD to process a spatial query and also satisfy the query-user parameters, answering the research question RQ 2. In order to provide a personalized result that also reflects the user's personal preference instead of only the query parameters, this chapter discusses models to personalize spatial information systems. The user models provide tools to distinguish users based on their needs and preferences, answering the research question RQ 3. In addition, the next challenge is to order the POIs in the rank considering the user preference and the query-parameters. Probabilistic functions have been used by Recommender Systems to model the user preference and rank POIs. For this reason, it is revised these probabilistic functions and other ranking strategies employed by different spatial queries help to address the research question RQ 1. The next chapter introduces the framework to improve the order of items in spatial keyword preference queries.

THE FRAMEWORK FOR IMPROVING SPATIAL KEYWORD QUERY APPLICATIONS

The goal of this chapter is to discuss and propose a framework to build location-based solutions. It has three modules that exploit the benefits of LOD, personalize the query results, and re-order the query rank. The first module enhances the textual description of features by accessing LOD repositories. Then, the second module re-orders the query results by applying query personalization with textual classifiers. The third module models the average user preference with the Pareto distribution, ranking POIs with our novel ranking function.

5.1 MODULES OVERVIEW

This D.Sc. thesis presents a framework composed of three modules to improve the searching and rank order of POIs presented to the user:

- **Module 1:** increase the description of features by extracting information from LOD repositories;
- **Module 2:** personalize the results retrieved by the query;
- **Module 3:** re-order the results considering a user average preference.

Figure 5.1 depicts an overview of the framework proposed by this thesis to automatically improve the query results generated by spatial keyword preference queries. In this figure, the components inside dotted lines describe module requirements while the numbers identify the modules. Initially, the user poses a query to describe his/her information need through query keywords. During the query processing, the first framework's module improves the description of features, exploiting LOD (described by module 1 in Figure 5.1).

Given the circumstance that traditional spatial datasets (e.g. Google Maps or OpenStreetMap) represent the objects using only its names as textual descriptions, module

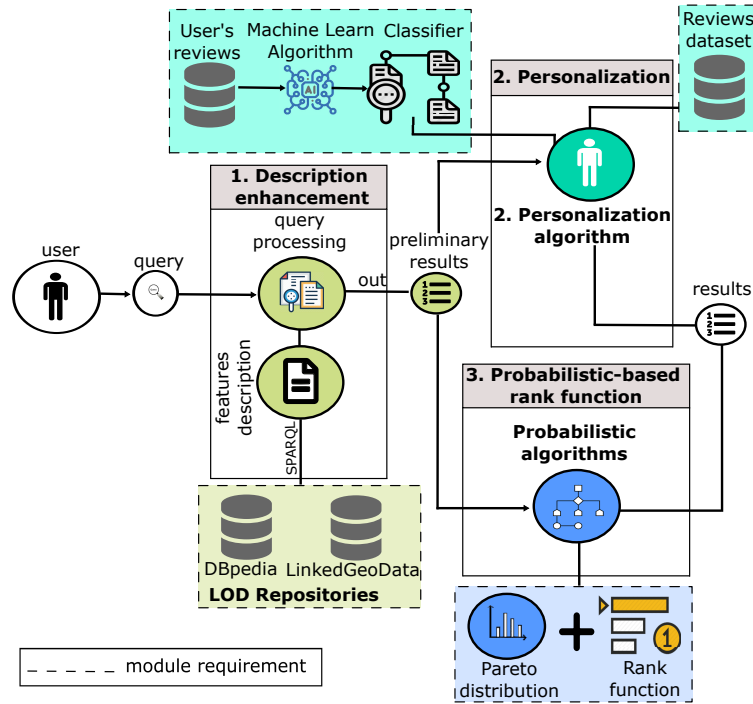


Figure 5.1: Overview of our approach to automatically improve query results.

1 enhances textual descriptions accessing different LOD repositories through SPARQL queries. This algorithm concatenates the object’s description in DBpedia (e.g. abstract) with its respective name in OpenStreetMap. In contrast, the traditional SKPQ uses a Spatial Inverted Index (S2I) (see Section 2.8.1) to index the textual dataset containing all textual descriptions needed instead of accessing the descriptions in LOD repositories.

After the textual description enhancement, the query processing continues until the rank is defined, generating a preliminary result set. Modules 2 and 3 can improve the preliminary results further. The personalization algorithm (module 2) re-orders the query results exploiting a user model generated by a textual classifier. The personalization algorithm re-evaluates the position of each POI in the rank considering the user model.

Module 3 considers the average user preference to visit POIs close to each other. We demonstrate that the Pareto distribution is a satisfactory function to model this preference. Thereby, we propose incorporating the Pareto distribution in the ranking function to re-order the preliminary results considering this particular user preference.

Consequently, this chapter presents the details of each module to improve Spatial Keyword Preference queries. It consists of the following sections: Section 5.2 presents the SKPQ enhancement with LOD; Sections 5.3 and 5.4 describe the personalization algorithm and the probabilistic algorithm. Finally, Section 5.5 concludes the chapter.

5.2 FEATURE DESCRIPTION ENHANCEMENT ALGORITHM (SKPQ-LD)

Figure 5.2 illustrates an overview of the feature description enhancement (SKPQ-LD) algorithm. This algorithm searches for relevant features in each POI neighborhood, and

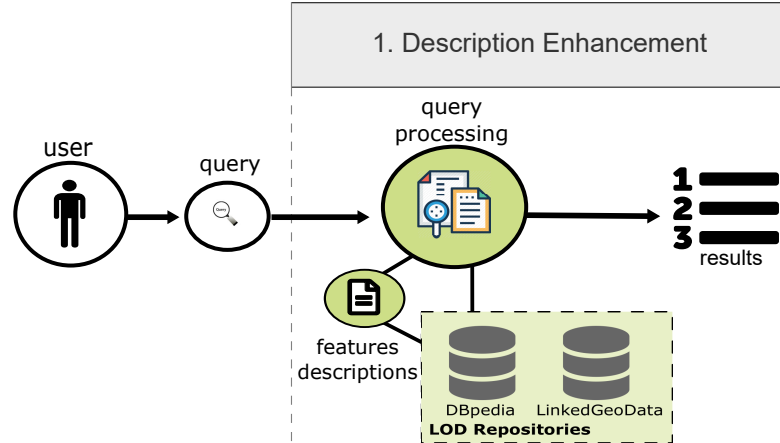


Figure 5.2: Overview of the textual enhancement algorithm by exploiting LOD.

it enhances the description of the retrieved features using LOD. The relevant feature has a description that shares words in common with the user query keywords. First, the user poses a query describing his/her information need by defining the query radius, the query keywords, and the number of expected results. For example, to search for POI near a jazz club, the user can define the query keywords “jazz club”, the query radius as 200m, and the number of expected results as ten. Then, the algorithm searches for features in the neighborhood of each POI. It enhances the features’ description and computes the textual relevance between the enhanced description and the query keywords. As a result, the algorithm returns the POIs that best satisfy the user information need. In other words, it returns a set of POIs with a textually relevant feature in their neighborhood. This module contributes to achieving the SO 2 and contributes to answering the RQ 1 and RQ 2.

In traditional SKPQ, the textual description of features is previously indexed using S2I. The indexing process has a high computational cost but enables 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. Consequently, 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 processes the Spatial Keyword Preference Query with LOD (SKPQ-LD) that searches for features in LOD repositories instead of the S2I. It receives as input the query $q = \{q.d, q.r, q.k\}$, 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 2-16). 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 by executing a SPARQL query (see Section 6.1.1.1). 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 ($\theta(f.d, q.d)$). We use cosine similarity because we want the term frequency to be determinant over the

Algorithm 1: Processing SKPQ-LD - the SKPQ that accesses LOD repositories.

Input: $q = (q.d, q.r, q.k)$

Output: Iterator over the elements in the Heap H that maintains the k best POIs

```

1  $H \leftarrow \emptyset$ 
2 for each  $p \in P$  do
3    $\tau(p, q) \leftarrow 0$ 
4    $iterator \leftarrow findObjectF(objectP).iterator()$ 
5   while  $iterator.hasNext()$  do
6      $f.d \leftarrow getAbstract(iterator.next())$ 
7      $\tau(f, q) = \theta(f.d, q.d)$ 
8      $\tau(p, q) = max\{\tau(f, q)\}$ 
9   end
10  if  $|H| < k$  OR  $\tau(p, q) > H.peekMin().score$  then
11     $H.add(p)$ 
12    if  $|H| > k$  then
13       $H.removeMin()$ 
14    end
15  end
16 end
17 return  $H.descendingIterator()$ 

```

document length (ZOBEL; MOFFAT, 2006). The method $getAbstract(iterator.next())$ (line 6) processes the SPARQL query described in Listing 6.2 to obtain the features' textual description. After computing the score of the feature f , the function $\tau(p, q) = max\{\tau(f, q)\}$ updates the score of p with the maximum feature's textual score $\tau(f, q)$ in the neighborhood 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 ($\tau(p, q) > H.peekMin().score$). 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 algorithm complexity is $O(|P| \cdot |F'|)$.

5.3 QUERY RESULT PERSONALIZATION ALGORITHM (P-SKPQ)

Consider a user that writes reviews about the locations he/she has visited. These reviews can describe the user's preference. This way, module 2 uses them to build a user profile. Each query user of the system is associated with a user profile composed of reviews made by them. Before the query processing, a textual classifier is trained by using the user

profile to learn his/her preferences. Figure 5.3 depicts this training stage that occurs before the query processing and the personalization algorithm that is executed after the query processing to re-order the query results. This module contributes to achieving the specific object SO 3 - Propose algorithms to personalize SKPQ, and contributes to answering the RQ 1, RQ 3, and RQ 4.

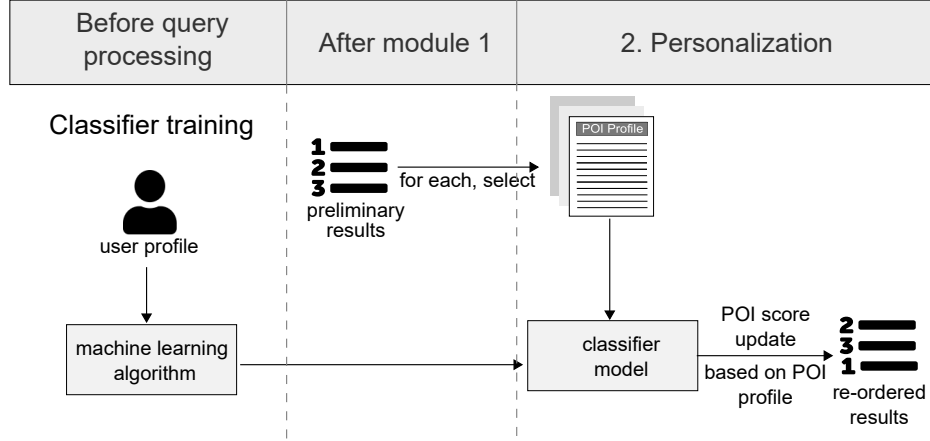


Figure 5.3: Overview of the personalization algorithm.

For each POI in the preliminary result set, the module 2 accesses its respective profile to update the POI score in the rank. The POI (e.g. a hotel) profile contains reviews of different users describing the POI. The classifier model is used to classify each review in the POI profile as good or bad to the query-user, represented by the integer values one and zero. Then, the algorithm sums each label generated by the classifier. This summation value represents the POI preference score considering the user preference learned by the classifier model. Therefore, the POI preference score reflects the query-user preference to that POI. The personalization algorithm (Algorithm 2) details this process.

After the SKPQ-LD is processed and the heap H with the preliminary results are generated, the personalization algorithm updates the score of each POI in H . For each $p \in H$, a set of reviews (REV) 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 $rev \in REV$ 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 rev , it receives a value of 1 (good); otherwise, it receives 0 (bad).

Each set of reviews REV contains a different number of reviews describing the POI. For this reason, we employ an accumulator c that has its value incremented when the review is classified as good or decreased when the review is classified as bad (lines 5-12). Then, the accumulator value is normalized as described in line 13. In the end, the POI score is updated (line 14) to reflect the user preference described in the user profile.

Considering c value as 1, the POI may go to the top of the rank. Before the personalization, the POI score $\tau(p, q)$ is composed only by the cosine similarity value representing the similarity between the query keywords and the feature's textual description. Since the cosine similarity value ranges from 0 to 1, $c = 1$ doubles the value of $\tau(p, q)$ when the

Algorithm 2: Personalization algorithm to re-order the query result.

Input: $H_k = \{p_1, p_2, \dots, p_k\}$
Output: Iterator over the elements in the Heap H_k in descending order

```

1 for each  $p \in H$  do
2    $REV \leftarrow getReviewSet(p)$ 
3    $\tau(p, q) \leftarrow$  POI score after query processing
4    $c \leftarrow 0$ 
5   for each  $rev \in REV$  do
6     if  $classify(rev) == 1$  then
7        $c++$ 
8     end
9     else
10       $c--$ 
11    end
12  end
13   $c \leftarrow \frac{c}{|REV|}$ 
14   $\tau(p, q) \leftarrow \tau(p, q) + c$ 
15 end
16 return  $H.descendingIterator()$ 

```

POI score is updated (line 14). As a result, the value of c changes the p rank position according to the user preference described in the user profile.

5.4 PROBABILITY-BASED RANKING FUNCTION

Module 3 consists of two algorithms to exploit the probability-based ranking function: the Probability-based Search Model (PSM) and the Probability-based Ranking Re-order (PRR). This section presents these algorithms for processing the SKPQ applying the novel ranking function to search for POIs or re-order the query result. Initially, a statistical analysis is conducted to verify the Pareto distribution suitability to describe the interest in POIs close to each other. Then, the traditional and the novel score function are defined. Finally, the algorithms that jointly exploit textual relevance and the user's implicit preference to produce the query results are described. This module contributes to achieving the SO 4 and contributes to answering the RQ 1, RQ 3, and RQ 4.

5.4.1 Data Analysis

The user movement exhibits structural patterns regarding geographical constraints (CHO; MYERS; LESKOVEC, 2011). Considering these patterns, studies have focused on building models of human movement with the intent to improve large scale systems, such as city and transportation planners or location-based recommenders (JIANG; FERREIRA; GONZALEZ, 2017; YANG et al., 2017; ZHANG et al., 2018). In order to verify if the

Pareto probability is a reasonable probabilistic function to estimate the user visiting probability, we perform a data analysis in the neighborhood of POIs (i.e. hotels) from four cities: Berlin, London, Los Angeles, and New York.

A widespread method to monitor human movement is the check-in. Check-in refers to record the time and location of a user during an event (e.g. visiting a restaurant). Zhang et al. (2018) perform data analysis on Foursquare check-in records from Los Angeles and San Diego. The authors demonstrate that the visiting probability decreases as the distance between the last place visited and the next increases. Since Foursquare API does not provide check-ins location anymore¹ due to privacy concerns, module 3 is inspired by the methodology of Zhang et al. (2018) to perform similar data analysis.

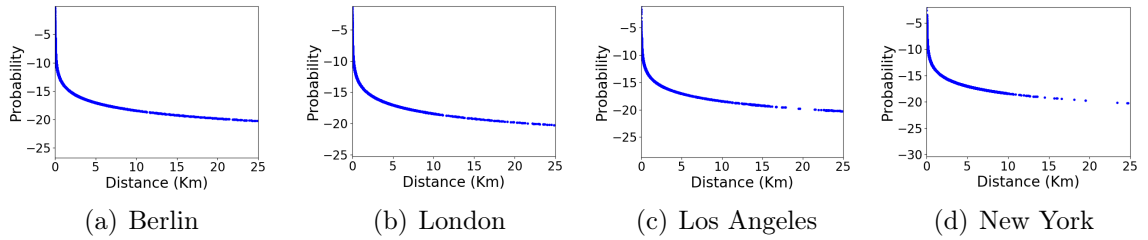


Figure 5.4: The distance distribution between the POI location and features in its spatial neighborhood.

The top- k spatial preference query user has the interest to find a POI nearby a specific feature (e.g. feature’s description matches with user keywords). Thereby, we calculate the distance between the POI and every feature in its spatial neighborhood (1 km radius). Following the methodology of Zhang et al. (2018), we apply the Pareto probability function to model the distance distribution. We observe in Figure 5.4, that the probability decreases as the distance between POI and feature increases. Therefore, the Pareto distribution can be employed to describe the human behavior of visiting locations close to each other. A similar observation is reported by Zhu et al. (2015).

The use of the Pareto distribution to describe an implicit user preference without relying on the personal data of the users avoids privacy concerns, not requiring the user to share his/her data to search for POIs.

5.4.2 The Ranking Function

Considering the analysis in Section 5.4.1, the assumption posed here is that the average user is not interested in features too far from the POI. We use the Pareto probability as a function to model this implicit geographical factor. Therefore, our ranking function incorporates the Pareto probability to rank the POIs.

Let P be the POIs dataset (P), where each POI $p \in P$ is represented by its spatial coordinates $p = (p.x, p.y)$. The SKPQ is denoted as $q = (q.d, q.r, q.k)$, where $q.d$ is the set of query keywords, $q.k$ is the number of expected results, and $q.r$ is the query

¹<<https://support.foursquare.com/hc/en-us/articles/201065830-Privacy-on-Foursquare>>

radius. Given a set of features F , in which each feature $f \in F$ is associated with spatial coordinates $f.x$ and $f.y$, and a textual description $f.d$. A query q returns a rank containing the top- k POIs $P_k = \{p_1, p_2, \dots, p_{q,k}\}$ with the highest scores $\tau(p_1, q) \geq \tau(p_2, q) \geq \dots \geq \tau(p_k, q)$. The score of a POI is determined by the highest textual relevance among all features that are in the POI's spatial neighborhood. The query radius $q.r$ defines the POI's spatial neighborhood. The SKPQ employs a similarity function to define the textual relevance between the query keywords $q.d$ and the features' description $f.d$. The score of a POI p for a given query q is defined in the following function:

$$\tau(p, q) = \max\{\theta(f.d, q.d) \mid f \in F : \text{dist}(p, f) \leq q.r\} \quad (5.1)$$

where $\theta(f.d, q.d)$ is the similarity between the feature description $f.d$ and the query keywords $q.d$. Consequently, a POI is part of the query rank P_k , if and only if exists at least one term in $q.d$ that is also in $f.d$. The Euclidean distance between the POI p and the feature f is denoted as $\text{dist}(p, f)$. Therefore, the traditional ranking function $\tau(p, q)$ only considers the features' textual relevance and its distance to the POI. Aiming to satisfy the user better, we propose to modify the ranking function to include a better representation of the user's geographical preference:

$$\tau^P(p, q) = \max\{\alpha \cdot \theta(f.d, q.d) + (1 - \alpha) \cdot Pr'(\text{dist}(p, f)) \mid f \in F : \text{dist}(p, f) \leq q.r\} \quad (5.2)$$

where for each $f \in F$ the distance to the POI have to satisfy the spatial selection criteria $\text{dist}(p, f) \leq r$. The function $Pr'(\text{dist}(p, f))$ is the normalized probability of a user visit the feature f given the distance to the POI p . $\theta(\cdot)$ returns values within the range $[0, 1]$, for this reason we normalize the value returned by $Pr'(\cdot)$ in the same range. Thereby, Pr' and θ have the same weight in the ranking function.

The *query preference parameter* α defines the importance of the textual relevance θ over the visiting probability Pr' into the ranking function $\tau^P(p, q)$ (ATTIQUE; KHAN; CHUNG, 2017; ROCHA-JUNIOR et al., 2011; SALGADO; CHEEMA; TANIAR, 2018). For example, $\alpha = 0.5$ means that the textual relevance and the visiting probability are equally important. In the following, we define the measures in more detail.

5.4.2.1 Textual relevance (θ) The textual relevance can be any function that returns the similarity between the query keywords $q.d$ and the feature description $f.d$ (ATTIQUE; KHAN; CHUNG, 2017; MANNING et al., 2012; ROCHA-JUNIOR et al., 2011). In this paper, we adopt the widely known cosine similarity between the weights vectors of the words in $q.d$ and $f.d$:

$$\theta(f.d, q.d) = \frac{\sum_{t \in q.d} w_{t,q.d} \cdot w_{t,f.d}}{|V_{q.d}| |V_{f.d}|} \quad (5.3)$$

where $w(\cdot)$ measures the weight of the term t in the query keywords $q.d$ or in the feature description $f.d$. The Euclidean norms of the weighted vectors are represented by $|V_{q.d}|$ and $|V_{f.d}|$. Other types of textual relevance measures such as Jaro-Winkler distance (COHEN; RAVIKUMAR; FIENBERG, 2003), Fuzzy score (ASTRAIN;

MENDÍVIL; GARITAGOITIA, 2006), or Okapi BM25 (MANNING et al., 2012) could also be employed by our ranking function.

5.4.2.2 Pareto probability (Pr) Considering the analysis in Section 5.4.1, we adopt the Pareto distribution to model the distance distribution between the POI and the features in its spatial neighborhood. Accordingly, the probability density function of the Pareto distribution is determined as follows:

$$f(x; \gamma, \beta) = \frac{\gamma\beta^\gamma}{x^{\gamma+1}} \quad (5.4)$$

where γ is the shape parameter, x is the distance of a feature to a POI ($dist(p, f)$), and β is the minimum value of x . Since β is the minimum distance x between the POI and the feature in its spatial neighborhood, β value can not be 0. Hence, β value is fixed $\beta = 1$. Thereby, Equation 5.4 can be written as follows:

$$f(dist(p, f); \gamma) = \frac{\gamma}{dist(p, f)^{\gamma+1}} \quad (5.5)$$

However, Equation 5.5 generates small probability values. For this reason, we adopt the natural logarithm of the probability density function to represent the values in the logarithmic scale. Therefore, the user visiting probability Pr is determined by the Equation 5.6:

$$Pr(dist(p, f)) = \ln \left(\frac{\gamma}{dist(p, f)^{\gamma+1}} \right) \quad (5.6)$$

Equation 5.7 normalizes the visiting probability, where min_p and max_p are the minimum and maximum user visiting probability of a feature f in the neighborhood of p .

$$Pr'(dist(p, f)) = \frac{Pr(dist(p, f)) - min_p}{max_p - min_p} \quad (5.7)$$

5.4.3 Probability-Based Search Model (PSM)

After the user describe his/her information need, the PSM algorithm searches for POIs that satisfy the query considering the novel ranking function proposed in this thesis. This ranking function employs the Pareto distribution to model the average user preference for places near to each other. Therefore, the novel ranking function considers the average user preference and also the textual relevance between the feature and the query keywords as described in Section 5.4.2. Figure 5.5 illustrates that the PSM algorithm also enhances the textual description of features as the Algorithm 1 employed to process the SKPQ-LD in Section 5.2. However, the SKPQ-LD algorithm only considers the textual relevance to rank the POIs. In the end, the PSM algorithm returns a set of POIs that satisfy the user need and also satisfy the average user preference described in the literature.

The SKPQ-LD requires finding all features in the spatial neighborhood of each POI and computing the textual relevance of each feature's description to the query keywords. The feature that obtains the highest score in the POI vicinity determines its score. We

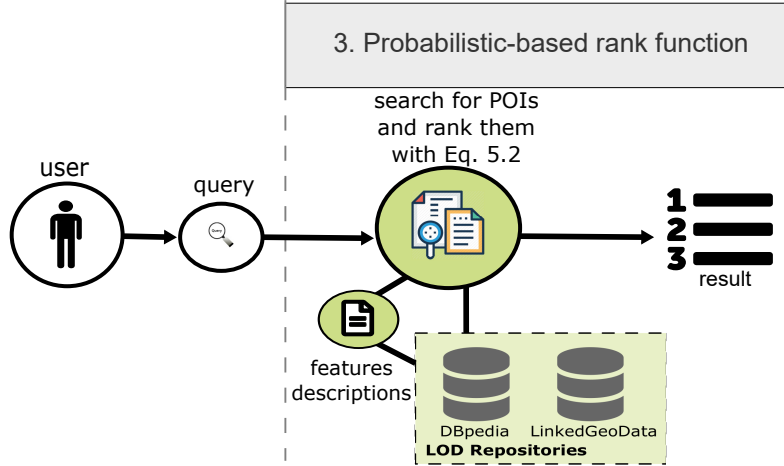


Figure 5.5: Overview of the PSM algorithm.

propose two new algorithms to process the query with the proposed ranking function: Algorithm 3 applies the ranking function during the search to select the best POIs, and Algorithm 4 applies the ranking function after the query processing to re-order the rank.

Algorithm 3 process the SKPQ adopting the ranking function described in Section 2.2. It receives as input the query $q = \{q.d, q.r, q.k\}$, and outputs an iterator containing the k best POIs. The algorithm computes the score of each POI $p \in P$ (lines 2-16). Initially, the score of p is zero (line 3). Then, all features f in the spatial vicinity of p are accessed (line 4). Given a set of features F' ($f \in F' : dist(p, f) \leq q.r$), we identify the maximum (max_p) and minimum (min_p) user visiting probability in the spatial vicinity of p . Then, the probability is normalized (line 6), and the score for each feature f in F' is calculated (lines 5-7). The ranking function determines the score of a feature $\tau^P(f, q)$ considering the visiting probability and textual relevance. The cosine similarity defines the textual relevance because the term frequency is determinant over the document length (ZOBEL; MOFFAT, 2006). After computing $\tau^P(f, q)$ for each feature in the vicinity of p , the score of p is updated using the maximum $\tau^P(f, q)$ value (line 8).

A POI p is added into the Heap H only if the score of p is higher than the lowest score among the objects currently stored in H (lines 9-14). If the size of H is larger than k , the POI with the smallest score in H is removed (lines 11-13). The algorithm returns a descending iterator to the k POIs with the highest scores stored in H (line 16).

5.4.4 Probability-Based Ranking Re-Order (PRR)

The PRR algorithm is illustrated by Figure 5.6. It processes the SKPQ-LD as described by the Algorithm 1 to search for the best POIs for the user. After the desired number of POIs are selected, the result is re-ordered by applying the novel ranking function proposed in this thesis (Equation 5.2). Therefore, the PRR algorithm has two steps: first, it finds the best POIs for the user considering only the query radius and the textual relevance between feature description and query keywords. Then, it re-orders the rank to boost the position of POIs that are close to its feature.

Algorithm 3: Probability-based Search Model (PSM) algorithm.

Input: $q = (q.d, q.r, q.k)$
Output: Iterator over the elements in the Heap H in descending order

```

1  $H \leftarrow \emptyset$ 
2 for each  $p \in P$  do
3    $\tau^P(p, q) = 0$ 
4    $F' \leftarrow \text{findFeatureSet}(p)$ 
5   for each  $f \in F'$  do
6      $\tau^P(f, q) = \alpha \cdot \theta(f.d, q.d) + (1 - \alpha) \cdot Pr'(dist(p, f))$ 
7   end
8    $\tau^P(p, q) = \max\{\tau^P(f, q)\}$ 
9   if  $\tau^P(p, q) > H.\text{peekMin}().score$  then
10     $H.add(p)$ 
11    if  $|H| > k$  then
12       $H.removeMin()$ 
13    end
14  end
15 end
16 return  $H.\text{descendingIterator}()$ 

```

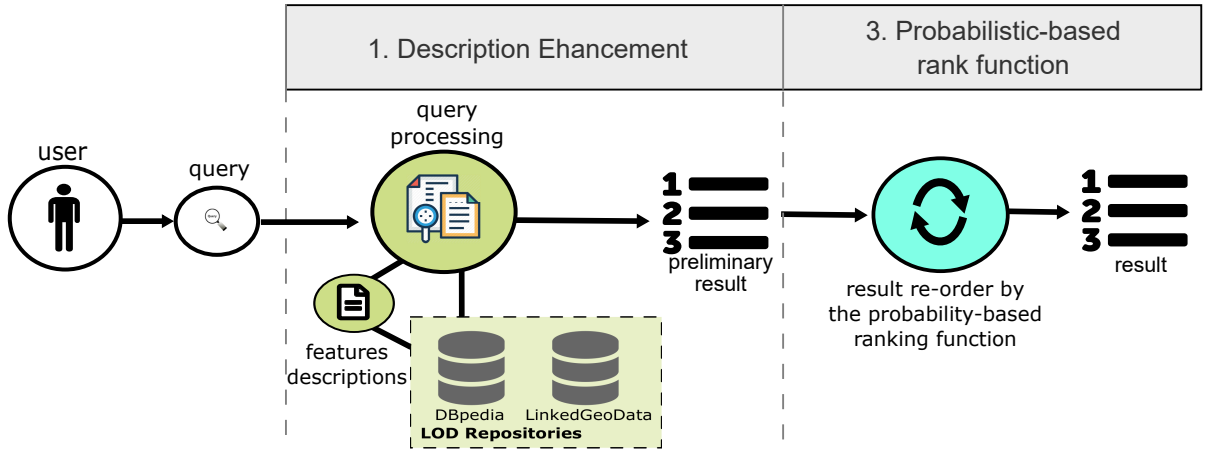


Figure 5.6: Overview of the PRR algorithm.

Query rank re-order is widely employed to improve rank accuracy considering user preferences (BOUIDGHAGHEN; TAMINE; BOUGHANEM, 2011; RATHOD; DESMUKH, 2017). Rank re-ordering consists of increasing the ranking position of results that satisfy the user preference. Therefore, it affects only the POIs in the query rank instead of selecting POIs in the search space.

After the query processing, the Heap H stores the result generated by the query. Thereby, every POI in H already has a score $\tau(p, q)$ that determines its rank position. Before re-ordering, the $\tau(p, q)$ value is defined by the maximum cosine similarity between

Algorithm 4: Probability-based Ranking Re-order (PRR) algorithm.

Input: $H_k = \{p_1, p_2, \dots, p_k\}$
Output: Iterator over the elements in the Heap H' in descending order

```

1  $H' \leftarrow \emptyset$ 
2 for each  $p \in H$  do
3    $Pr'(dist(p, f)) = (Pr(dist(p, f)) - min_p) / (max_p - min_p)$ 
4    $\tau^P(p, q) = (\alpha \cdot \tau(p, q)) + ((1 - \alpha) \cdot Pr'(dist(p, f)))$ 
5    $H'.add(p)$ 
6 end
7 return  $H'.descendingIterator()$ 

```

the query keywords and a feature description in the POI's neighborhood (Equation 5.1). The distance between the POI and its most textually relevant feature is calculated to obtain the user visiting probability (Equation 5.6) and re-order the rank. This process repeats for each POI in the rank to identify the maximum and minimum visiting probability.

In a sequence, the PRR algorithm (Algorithm 4) updates the score of each POI in the query rank. For each POI in H , the normalized Pareto distribution is calculated (line 3). Then, the POI's score $\tau^P(p, q)$ is updated considering the user visiting probability Pr' and its previously defined score $\tau(p, q)$ (line 4). Thereby, the ranking function described in line 4 increases or decreases the rank position of the POI based on its distance to the feature. Another Heap H' stores the re-ordered rank (line 5). The Pareto distribution describes the user preference for features close to the POI, as described in Section 5.4.1. Therefore, Algorithm 4 explores rank re-ordering to create a new query rank increasing the rank position of POIs that better satisfy the user's spatial preference.

5.5 SUMMARY

This chapter presented solutions to improve the SKPQ. This approach integrates a textual enhancement technique with query personalization and a novel rank based on a probabilistic function. The textual enhancement integrates data from different LOD repositories to describe POIs. The query personalization technique employs textual classifiers to re-order the query result. Another option to re-order the query results is presented by the algorithms that exploit the probability-based rank function. The algorithms described in this chapter answers the research questions RQ 1, RQ 2, and RQ 3. The next chapter presents the experimental evaluation conducted to assess each solution.

PART IV

EVALUATION

EXPERIMENTAL EVALUATION

This chapter presents the results obtained while assessing the modules and discusses their implications and limitations. A total of seven evaluations took place in order to assess each approach individually, considering the proper metrics and query parameters to evaluate each one. This chapter consists of the following sections:

- **Section 6.1** presents two experiments to evaluate the textual description enhancement. First, we compare the quality of the query results generated by SKPQ and SKPQ-LD then we evaluate if the proposal can select better features in the neighborhood of a POI than the traditional query;
- **Section 6.2** describes the query personalization results. We compare different classification algorithms to model the user preference considering reviews. Then, we compare the personalized SKPQ with variations of it to understand the impact of the result's personalization in this context. The queries are executed on two different datasets, varying the number of POIs in the result while using random and frequent keywords in each dataset;
- **Section 6.3** reports the results obtained by the probability-based ranking function. We vary the number of POIs in the query result and the number of keywords in the query to evaluate the query performance using our novel ranking function;
- **Section 6.4** describes the COVID-19 Geo-monitor use case developed to monitor the virus spread in a particular city. It processes the SKPQ considering infected patients as features to identify places whose neighborhood has a high rate of contamination. Also, it associates a patient to the nearest basic health unit;
- **Section 6.5** details the insights provided by a specialist about the use case COVID-19 Geo-monitor. We report the ratings each specialist gave to each function of the application and discuss their feedback about limitations and improvements;
- **Section 6.6** concludes the chapter.

6.1 MODULE 1 EVALUATION: FEATURES' DESCRIPTION ENHANCEMENT

The experiments to evaluate the textual description enhancement are performed in two ways, each with a unique methodology. In the first experiment, the users' ratings are extracted from Google Maps to evaluate the queries result. In the second experiment, the users' ratings were extracted from TripAdvisor¹ as described in Section 6.1.2. These two experiments are described in the following.

6.1.1 Experiment Setup

DBpedia and LinkedGeoData are accessed through the local repository or by the Snorql endpoint (see Section 6.1.2). All experiments are executed on the same computer with an Intel processor of 1.8 GHz (model i3-3217U) and 8 GB of RAM. To process the traditional SKPQ, we use the OpenStreetMap dataset indexed using the S2I. In contrast, to process the SKPQ-LD, we use the DBpedia dataset merged with OpenStreetMap dataset using SPARQL queries as discussed in Section 6.1.1.1. The code is developed in Java, using the Apache Jena² framework to access the LOD repositories. The code, and all requirements to execute it, are available at <<https://github.com/JoaoAlmeida/Enhancing-SKPQ>>.

The experiments follow two methodologies to evaluate the SKPQ-LD: using ratings obtained from Google Places API; and relevance judgments obtained from TripAdvisor. In Experiment 1, we apply the first methodology, where SKPQ and SKPQ-LD are 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 are selected randomly. The query results are evaluated using NDCG. The list of frequent terms is obtained from S2I³ and random queries keywords are obtained without repetition from a set of 1906 terms extracted from the OpenStreetMap dataset. "chili" and "sunset" are examples of random keywords in this experimental evaluation. Moreover, we use the object rate from Google Places API to determine the ideal ranking.

In Experiment 2, we apply the second methodology, where SKPQ and SKPQ-LD are executed using query keywords described in the OpinRank dataset. This dataset contains full reviews of hotels collected from Tripadvisor and their corresponding aspect ratings as described in Section 6.1.2. We use the queries related to each aspect as query keywords to 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 is used to process the traditional SKPQ and the SKPQ-LD. Both queries have the same parameters and use cosine similarity to evaluate the textual relevance between query keywords and the feature's description. We compare the ranks generated by these two queries to understand how LOD affects the object retrieval by the query. Experiments 1 and 2 contribute to answering the RQ 2 and validate the SO 2.

¹<<https://www.tripadvisor.com.br/>>

²<<https://jena.apache.org/>>

³Implementation available at XXL Library

6.1.1.1 SPARQL queries Once a feature f is found in the spatial vicinity of one POI p ($dist(p, f) \leq r$), its abstract and comment properties values are accessed from DBpedia using SPARQL. This abstract represents the textual description $f.D$. The textual score of f is computed using the cosine similarity function between the query keywords and the abstract $f.D$ as discussed in Section 5.2. In contrast, in the 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.

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

Listing 6.1: SPARQL query to find features that satisfies the spatial selection criteria.

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

Listing 6.2: SPARQL query to obtain textual description for one feature.

Listing 6.1 describes the SPARQL query used to search for features in the spatial vicinity of a POI, where *objectURI* represents the URI to access a POI. Similarly, the Listing 6.2 depicts the SPARQL query used to obtain the features' textual description $f.D$, where *referenceObjectURI* represents the URI to a feature.

6.1.2 Datasets

In this experimental evaluation, we use three data sources to process the SKPQ. The OpenStreetMap⁴ is used to process SKPQ and, DBpedia and LinkedGeoData are used to process SKPQ-LD. Additionally, two publicly available data sources are used to evaluate the query results: the Google Maps and the OpinRank dataset.

Extracts are pieces of OpenStreetMap data pruned at the region of individual continents, countries, or metropolitan areas. Mapzen⁵ maintains updated extracts for many cities. Mapzen is used to obtain OpenStreetMap data from Dubai, processing this data to extract only spatio-textual objects. The set of POIs P is composed of POIs whose category in the Dubai dataset is "hotel", while the set of features F is composed of the other spatio-textual objects. The Dubai dataset generated 162 POIs, 2243 features, 1906 unique terms, and 12256 terms in total as described in Table 6.1.

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

⁴<http://www.osm.org>

⁵<https://mapzen.com/data/metro-extracts/>

Table 6.1: Characteristics of the Dubai dataset obtained from Mapzen. The number of POIs $|P|$, the number of features $|F|$, the number of unique terms in the dataset, and the total number of terms.

Dataset	$ P $	$ F $	No. of unique terms	Total number of terms
Dubai	162	2243	1906	12256

To process SKPQ-LD, LinkedGeoData is accessed through SPARQL queries to obtain a set of POIs P equivalent to the one obtained from Mapzen, as illustrated by Listing 6.3. This SPARQL query returns a list of objects with the same name as the one stored at Mapzen but with 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 6.3: SPARQL query to obtain the points of interest to process SKPQ-LD.

In order to enrich the object’s textual description from LinkedGeoData, we use data 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 LinkedGeoData, we concatenate the text obtained in both data sources. The textual description $f.D$ obtained from DBpedia is composed by *rdfs:comment* and *dbo:abstract* predicates. For example, the Hotel Danieli from Venice is described as “(tourism) (hotel) Danieli” in OpenStreetMap (OSM). While in DBpedia, the same hotel is described as “Hotel Danieli, formerly Palazzo Dandolo, is a five-star palatial hotel in Venice, Italy. (..)”⁶. The hotel description in DBpedia is much wider than the OpenStreetMap description, with 58 more words.

We call “textual enhancement” this text concatenation from different sources. The enhanced text is stored in a local repository, consequently, the query does not have to access the online data source every time the query is executed in the experiments. In summary, both DBpedia and LinkedGeoData have public access. We accessed the text data from their respective endpoints, storing the retrieved data in a local repository. When the query searches for the textual description of an object, it first searches in the local repository. If the search fails, it looks for the information in the endpoints.

6.1.2.1 Ground-truth Dataset for Experiment 1 Besides the data sources used to process the SKPQ and SKPQ-LD, we used the Google Maps data and the OpinRank

⁶Full description can be accessed at <http://dbpedia.org/page/Hotel_Danieli>

Table 6.2: Example of information available in OpinRank dataset related to the query “great location”.

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

dataset to evaluate the queries. The Google Maps data is accessed through the Google Places API. This dataset contains POIs 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. The class RatingExtractor from our framework implements the users’ rating extraction from Google Maps.

6.1.2.2 Ground-truth 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 (2011) manually created textual queries related to each aspect rating. These queries were based on real queries made by users in popular search engines, thereby 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 6.2. 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 6.2. A full copy of the user queries and user ratings related to the aspect rating “room” is in Appendix B

6.1.3 Metrics

The metrics employed in the evaluation are Discounted Cumulative Gain (DCG), Normalized Discounted 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 IR, 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 6.3. 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 6.1 and 6.2, respectively. We denote top-k items by $P_k = \{p_1, p_2, \dots, p_k\}$, 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

$$DCG@k = \sum_{i=1}^{|P_k|} \frac{rel_i}{\log_2(i+1)} \quad (6.1)$$

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

$$IDCG = \max(DCG@k) \quad (6.2)$$

NDCG@k is calculated as

$$NDCG@k = \frac{DCG@k}{IDCG} \quad (6.3)$$

6.1.4 Experiment 1: Evaluating Query Results

To understand the ranking quality of both SKPQ and SKPQ-LD, we compare the NDCG values obtained when using random keywords and frequent keywords. Figure 6.1 reports the arithmetic mean of NDCG@k (k=5, 10, 15, 20) generated by the execution of twenty queries with distinct keywords. The arithmetic mean values are reported on the vertical axis. Figures 6.1 (b) and 6.1 (d) illustrate that SKPQ-LD improves the ranking quality when using random keywords; otherwise, the quality is roughly the same.

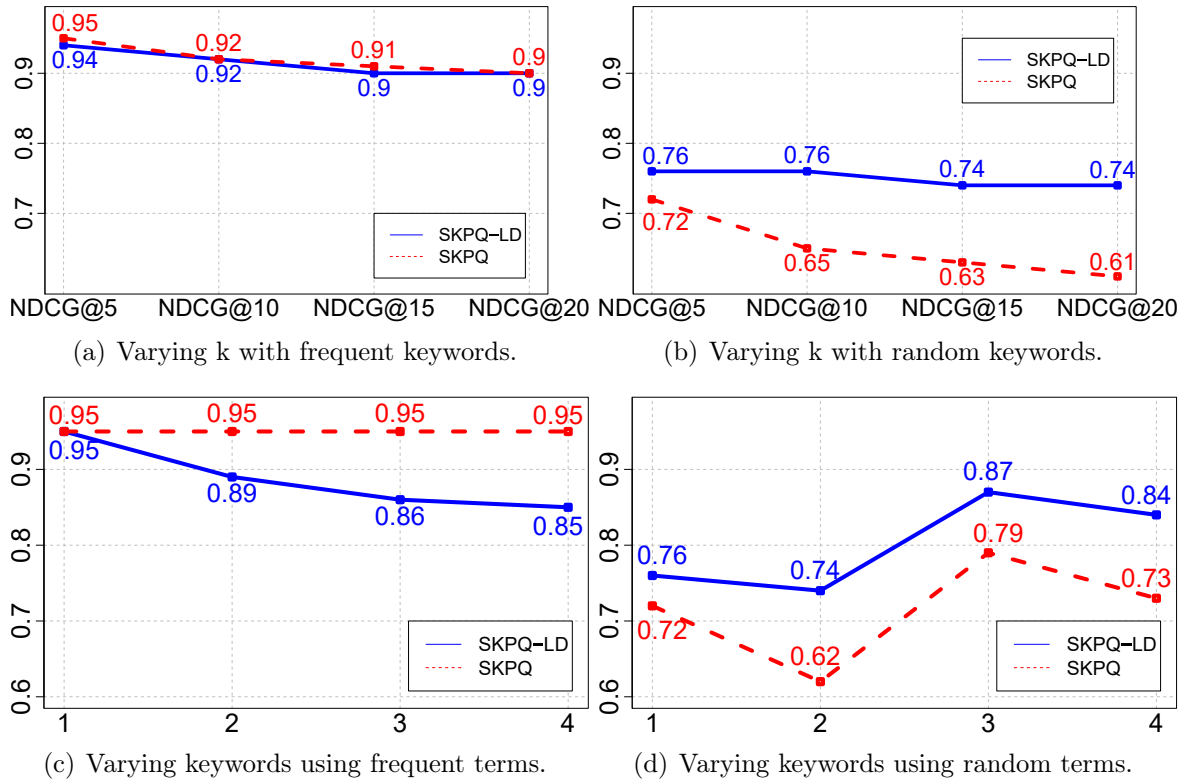


Figure 6.1: Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size (k).

It is noticeable that we obtain satisfactory results with SKPQ using frequent keywords. Since the keyword is present in many objects, there is no problem with SKPQ identifying the object that has textual relevance to the query keyword. In this scenario, the objects retrieved by SKPQ have a small textual description, but they have a high probability to match with the query keyword. In addition, the SKPQ accesses more objects because OpenStreetMap contains more spatial objects representing the city of Dubai than LinkedGeoData. Therefore, SKPQ counts on a good enough textual description and a larger amount of objects. These two factors lead to a better evaluation result to SKPQ when using frequent keywords. Nevertheless, the SKPQ-LD obtains results nearly as good as SKPQ, with a difference of only 0.1 between the NDCG values.

Figures 6.1(c) and 6.1(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 6.1(c) uses the ten most frequent terms in the dataset as query keywords. We combine the terms (without repetition) to build query keywords with two terms or more. For example, “chili” and “sunset” are combined to create the query keyword “chili sunset”.

As it can be seen in Figure 6.1(c), even after adding three more keywords, the results obtained in SKPQ do not change. However, SKPQ-LD is more influenced by the increase in the number of query keywords than SKPQ. As observed in Figure 6.1, 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 6.1(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 in 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, we employ our approach to enrich the textual description of objects accessed by the top- k Range Query (RQ) (CAO et al., 2012) and evaluate the results obtained. We call RQ-LD the RQ that uses our textual enhancement technique. Thereby, the RQ results are compared against the RQ-LD results. Given a spatial area and the query keyword, the RQ returns k objects in the given area that are textually relevant to the query keyword. All RQs use identical query keywords and radius as SKPQ. Moreover, the query location is randomly selected inside the Dubai dataset. The radius of 200 meters defines the spatial neighborhood of the RQ location. RQ is similar to SKPQ because it uses query keywords and searches for spatial objects inside an area defined by a radius. However, SKPQ searches for objects in the area around every POI in the dataset. While RQ only searches for objects in the area around the query location.

It can be seen in Figure 6.2 that our approach improves the RQ result set when using frequent keywords instead of random keywords. The RQ-LD behavior is the inverse of the SKPQ-LD behavior that presents better results when using random keywords. The RQ looks for all k objects in a small spatial area ($radius = 200m$) while SKPQ looks for

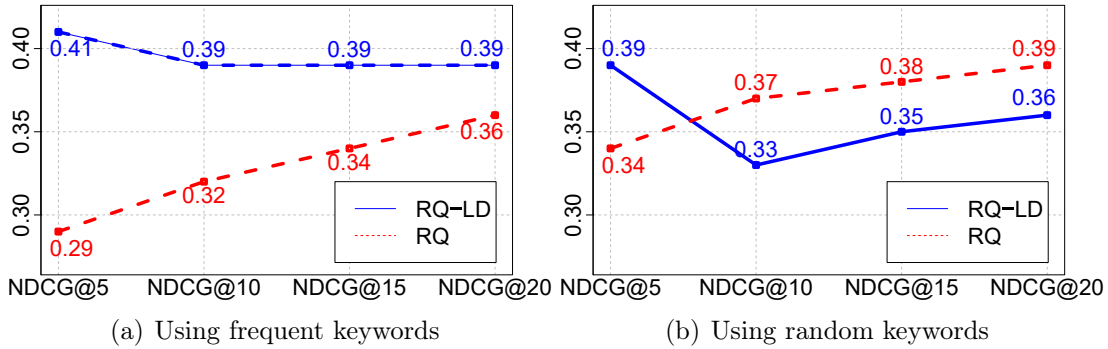


Figure 6.2: Results obtained with RQ and RQ-LD.

objects in the neighborhood of many points of interest. Each POI’s neighborhood has the same size as the search space visited by RQ (200 m). This contrast in search space size results in a more challenging effort to RQ build a quality rank for the given area; because there are fewer objects to verify. This can be verified by observing the much lower NDCG values obtained with RQ. While SKPQ obtains 0.61 in its worst case, RQ obtains 0.41 as its best case. We believe that the amount of objects to verify is the main reason for the lower NDCGs values depicted in Figure 6.2 than the ones in Figure 6.1.

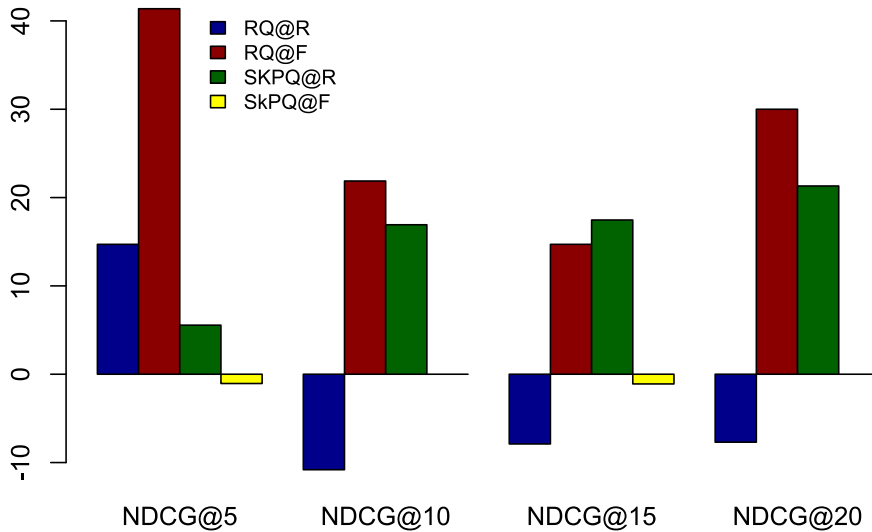


Figure 6.3: Relative NDCG improvements.

Figure 6.3 illustrates the relative NDCG improvement (SONG et al., 2016) of the proposed approach e_{pro} over respective baseline model e_{other} , further measured as

$$(e_{pro} - e_{other})/e_{other} \times 100 \quad (6.4)$$

Figure 6.3 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 by 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 evaluated if our approach improves the query result. Using random keywords, we conclude that 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 similar to the one obtained by the SKPQ. Therefore, our approach improves the result using random keywords; and does not impose a high penalty on the quality of the query result using frequent keywords.

6.1.5 Experiment 2: Evaluating Feature Selection

In Experiment 2, the queries in the OpinRank dataset are used to evaluate the feature selection in SKPQ and SKPQ-LD. We want to investigate the quality of POIs retrieved by SKPQ and SKPQ-LD, considering real user queries. Since the OpinRank dataset contains only hotel reviews, we restrict our feature dataset to hotels. All hotels accessed in this experiment are located in Dubai.

Given the query keywords, the SKPQ returns a list of points of interest whose locations 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 result is empty.

The OpinRank dataset offers five textual queries for each aspect rating (a total of 25 queries). These textual queries are used as query keywords in SKPQ. However, SKPQ did not find any feature whose textual description matched with the query keywords. The description accessed by SKPQ is too short and can not describe the feature as needed. Therefore, SKPQ is unable to retrieve relevant results to the OpinRank's queries. Notwithstanding, the SKPQ-LD is able to find textually relevant features. From 25 queries, SKPQ-LD can find relevant features in 15 (equals to 60% of all executed queries). Considering $k = 5$ and 25 as the number of executed queries, the MAP score obtained by SKPQ-LD is 0.46.

Among the fifteen relevant query results obtained by SKPQ-LD, we could extract the aspect rating value of few features. Many times, the hotel name in the OpinRank dataset was not found in DBpedia or OpenStreetMap. Hence, when SKPQ or SKPQ-LD retrieves a hotel whose name 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 query examples that did not return any relevant objects to the user. Therefore, the features' textual description accessed by 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 6.4 reports the NDCG values of the query results obtained with these query keywords.

With the textual enhancement of objects' description, SKPQ-LD was able to select

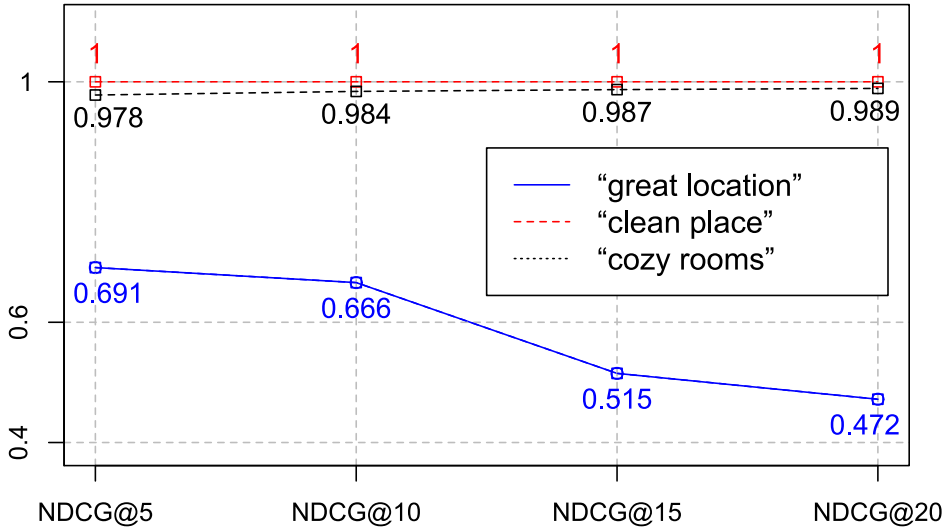


Figure 6.4: SKPQ-LD evaluation using OpinRank.

more objects that satisfy the user’s need than SKPQ. Accordingly to the obtained NDCG values in Figure 6.4, SKPQ-LD selected features of good quality. Since the query results have high aspect rating values, we can assume that SKPQ-LD is able to find satisfactory POIs for the user. For the query “clean place” for example, SKPQ-LD is able to find features that are evaluated by real users as a clean hotel.

The OpinRank dataset contains other queries created by combining the queries illustrated in Figure 6.4. For example, the combination of “great location” with “clean place” generates “great location clean place”. Nevertheless, these query combinations lead to results similar to the ones depicted in Figure 6.4. 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 is unable to find any relevant POIs to the real-users queries.

6.1.6 Discussion: Datasets Characteristics and Features Description

We ran the SKPQ varying query keywords to extend our analysis in order to understand the difference between textual descriptions from DBpedia and OpenStreetMap. Table 6.3 is an experiment result using Venice hotels as points of interest (P) and “church” as a query keyword. The first column of Table 6.3 presents the Point of Interest (POI) textual description, the second column has the POI score using SKPQ, and the third column presents the POI’s score using SKPQ-LD. In order to find features that satisfy the spatial selection criteria and to assign a score to a POI, the *geo:geometry* property has to exist in the LOD object. For this reason, the POIs “Palazzo Ferro Fini” and “Splendid Venice” has no score in Table 6.3.

Some cities (e.g. Venice) contain few spatial objects represented at DBpedia. This LOD database contains only five hotels in Venice against 488 registered in OpenStreetMap. Despite the considerable object number, the textual description in OpenStreetMap has

low quality. While a typical textual description in DBpedia has around 60 terms, the textual description for the same object has only two terms in OpenStreetMap. The poor textual description leads the SKPQ to misjudge the evaluation of some POIs. Given the query keyword “church”, objects “Hotel Cipriani” and “Hotel Danieli” have features in their spatial neighborhood that are textually relevant to the query keyword, but SKPQ fails to identify them because of poor textual description. SKPQ-LD did find these objects and was able to retrieve “Hotel Cipriani” and “Hotel Danieli”. For the same reason, SKPQ did not find relevant POIs in Experiment 2 described in Subsection 6.1.5.

Table 6.3: Score of object p in traditional SKPQ compared with the score generated by SKPQ-LD, using hotels from Venice.

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 <i>geo:geometry</i> property
Splendid Venice	0	no <i>geo:geometry</i> property

Table 6.4: Score of object p in traditional SKPQ compared with the score generated by SKPQ-LD, using hotels from São Paulo.

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

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 6.4. The column names in Table 6.4 have the same meaning as the column names in Table 6.3. This time we have no problem finding the *geo:geometry* property, but SKPQ still has issues retrieving POIs. SKPQ still returns more objects with a score of zero than SKPQ-LD. These results endorse the improvement obtained by our approach when using random query keywords because “church” is a random query keyword in this dataset.

As illustrated in Table 6.4, the score of the object “San Michel Hotel” when retrieved by SKPQ is higher than its score when retrieved by SKPQ-LD. 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 in SKPQ than SKPQ-LD. Therefore the term impact in SKPQ's document is more evident when the term exists in the document.

In summary, the results achieved by module 1 indicate that it is possible to exploit LOD to improve the SKPQ as intended in SO 2. The number of words describing the POI can determine the query capability in finding the POIs that best satisfy the user, answering the RQ 2. We observe improvement, mainly, when using random query keywords. Module 1 also contributes to answering the RQ 5, by evaluating the queries considering the user rating and queries keywords from real users.

6.1.7 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, "Splendid Venice" (presented at Table 6.3) does not have the *geo:geometry* property hindering the textual description access by spatial queries.

Zarrinkalam and Kahani (2012) describe an enrichment approach using LOD to improve the textual description of articles citations. Accordingly to the authors, "the Linked Data driven enrichment process has improved the quality of recommendations but it isn't as much as expected" because "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 are 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. Thereby, it can search for more restaurants that satisfy the user's need.

The world of Linked Data poses many challenges, as described by 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.

6.2 MODULE 2 EVALUATION: QUERY PERSONALIZATION

The goal of this evaluation is to verify when reviews improve spatial keyword preference queries accuracy. In this section, we present our datasets, our methodologies, and the results obtained during the experimental evaluation. This evaluation covers the RQ 1, RQ 3, RQ 4, and RQ 5. It also contributes to the SO 3.

6.2.1 Experiment Setup

The DBpedia and LinkedGeoData are accessed through an HTTP endpoint. All experiments are executed in the same computer using an Intel processor of 1.8 GHz (model i3-3217U) and 8 GB of RAM. We use Java, the Apache Jena framework, and Weka⁷ to develop the code. The queries are processed using dataset frequent and random words as query keywords. The list of frequent words are obtained from the textual descriptions stored in S2I⁸ while the random words are obtained without repetition from the set of unique terms extracted from the POIs dataset described in Subsection 6.2.2. “chili” and “sunset” are examples of random keywords used in this thesis. The code to reproduce this experiment is available at <<https://github.com/JoaoAlmeida/Enhancing-SKPQ>>.

In order to evaluate the query personalization approach, we build the 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 classification model.

6.2.2 Datasets

Similar to the evaluation of the textual enhancement (module 1), three data sources are employed to process the query personalization: OpenStreetMap, DBpedia, and LinkedGeoData. Each data source is used to create the set of points of interest (1) and features (2). Additionally, the OpinRank dataset is used to evaluate the obtained query results (see Section 6.1.2.2 for details). The description of each data source is detailed in sequence:

1. **Points of interest:** extracts are pieces of OpenStreetMap data pruned at the region of individual continents, countries, or metropolitan areas. Similar to the process applied to evaluate SKPQ-LD (Section 6.1.2), we use Mapzen to obtain an extract from Dubai, then we process the data to select only spatio-textual objects. The set of POIs P is composed of POIs whose category in OpenStreetMap dataset is “hotel”. The *extract* representing Dubai generated 162 hotels, 1 906 unique terms, and 12 256 terms in total used to describe the hotels. Also, we use the dataset described by (ALMEIDA; ROCHA-JUNIOR, 2016) containing POIs in London. They obtained the London dataset using the same method as the one described in this section to acquire the Dubai dataset (ALMEIDA; ROCHA-JUNIOR, 2016). The *extract* representing London has 672 hotels, 56 569 unique terms, 1 198 649 terms in total. The POIs are stored using S2I - an index for efficient query processing.

⁷<<https://www.cs.waikato.ac.nz/ml/weka/>>

⁸Implementation available at XXL Library

2. **Features:** during the SKPQ-LD process, the query searches in LinkedGeodata and DBpedia for features that are close to each POI. Both LinkedGeodata and DBpedia are Linked Open Data repositories. Using an HTTP endpoint, we access LinkedGeoData to search for features that satisfy the spatial selection criterion. When the desired feature is found, we obtain its textual description and apply our textual enhancement approach. Thereby, DBpedia is accessed to improve the textual description of this feature, concatenating the text obtained in both datasets. A detailed description of this process is described in Section 6.1.2.

6.2.3 Metrics

The metric employed to assess the query results is the Normalized Discounted Cumulative Gain (NDCG). This metric is described in Section 6.1.3.

6.2.4 User profiles

User profiles employ past reviews of the user to describe his/her preference. It is possible to indicate the best item for the user manipulating this preference description. As discussed in Section 6.1.2.2, each hotel has five aspect ratings in the Opinrank dataset: cleanliness, room, service, location, and value. One user profile is built for each aspect rating to simulate a real user profile. Each profile consists of twenty user reviews and their respective label: 0 for a bad review (negative class) and 1 for a good review (positive class). For instance, when building the *service* aspect rating related profile, a specialist selected ten reviews about hotels with the highest *service* aspect rating value and another ten about hotels with the lowest. Thereby, the *service* related profile represents a user who has visited hotels with good and bad services and commented about them on TripAdvisor.

Besides, the user profiles are location-aware - this means that all reviews used to describe the user preference in the profile are filtered. Hence, the user preference is described only by user's reviews about POIs in the same city the user is interested in conduct a search. For example, a user interested in finding a POI in Dubai has a user profile containing only reviews made about other POIs in Dubai.

Table 6.5 illustrates a user profile where each line displays the label followed by the user review. Usually, reviews have a larger number of characters than the ones described in Table 6.5. All profiles are available at <http://tiny.cc/i3babz>, and a copy of the *room* aspect rating related profile is on the Appendix A.

The review text is filtered using the StringToWordVector method (ADELEKE et al., 2018). It converts the string into a vector containing a set of attributes representing term frequency. No further text preprocessing is applied. The vectorized profile is used to train the machine learning algorithm. It generates the model to classify the reviews from hotels the user has not visited yet, as described in Section 5.3.

Table 6.5: Example of the user profile related to service aspect rating.

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.

6.2.5 Classification model

Figure 6.5 describes the process employed to generate the classification model. In the training stage, the user's reviews about hotels he/she has visited coupled with their respective labels are the input to the machine learning algorithm. The text of each user review is vectorized by using the StringToWordVector method. Thereby, instead of text, we use a vector with word counts representing the user review.

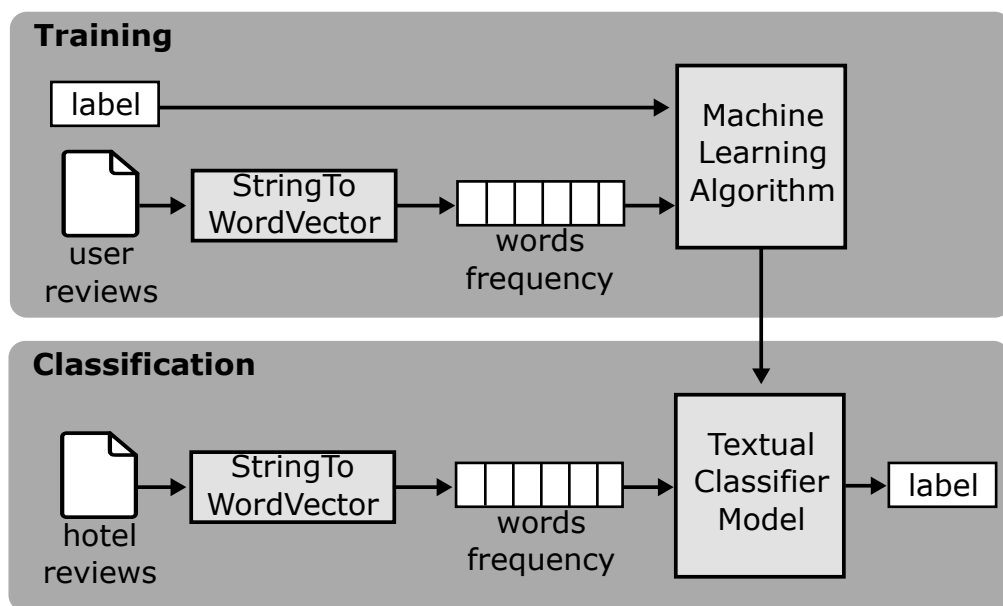


Figure 6.5: Classification model to learn user preference based on his/her past reviews. Source: Adapted from Bird, Klein and Loper (2019).

In the preliminary results (described in Figure 5.1), each POI corresponds to a hotel. Each hotel in the rank is associated with a set of user reviews made by users who have visited the hotel. Similar to the training stage, each review is vectorized by the StringToWordVector method. Then, the word frequency vector is classified by the model. Thereby, the model labels the hotel review considering the user preference described by his/her reviews in the training stage.

6.2.6 Choosing the Classifier

An experiment is conducted to evaluate and choose the best classifier to personalize the query results. First, classifiers are trained and then compared using classification accuracy, and F-measure. In the sequence, we apply the kappa coefficient to measure the agreement between the ideal results with the results generated by the classifier.

There are five user profiles simulating users with different preferences (cleanliness, room, service, location, and value). Each profile is employed to train and test classifiers using different algorithms. We take some traditional classification algorithms from Weka: Naive Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), Logistic Regression (LR), and Decision Tree (DT). All classifiers are trained using the 5-fold cross-validation procedure, in which all algorithms are subjected to the same number of folds.

The classification accuracy (Equation 6.5) is defined as the ratio of the number of correct classifications versus the total number of classifications. F-measure (Equation 6.8), also called F-score, is the harmonic mean of precision and recall of the positive review class. Precision and recall are defined by equations 6.7 and 6.6, respectively. All measures range from 0 to 1.0, where 1.0 is the best classifier performance.

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of classifications made}} \quad (6.5)$$

$$Recall = \frac{\text{Number of correct positive classifications}}{\text{Number of positive examples}} \quad (6.6)$$

$$Precision = \frac{\text{Number of correct positive classifications}}{\text{Number of positive classifications}} \quad (6.7)$$

$$F - \text{measure} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6.8)$$

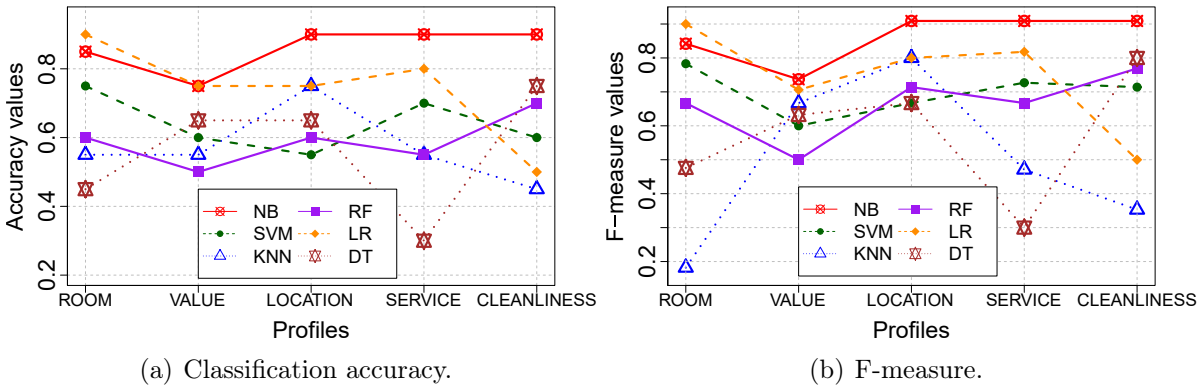


Figure 6.6: Classifiers' performance evaluation varying training data and algorithms.

The best KNN classifier is the one using the location profile that achieves an accuracy of 0.75, as illustrated in Figure 6.6(a). Likewise, the best SVM classifier (room profile) achieves the same classification accuracy such as the best KNN classifier. Nevertheless, NB classifiers performed better than the ones trained using SVM or KNN in all scenarios. In fact, NB classifiers trained using the service, location, and cleanliness profiles present the best classification accuracy values between all classifiers trained with these profiles. The only classifier that achieves similar results is the LR classifier trained with the room profile, obtaining a classification accuracy of 0.9.

Figure 6.6(b) depicts the F-measure value for each trained classifier. Likewise the classification accuracy experiment, NB classifiers present the best F-measure values, reinforcing that NB algorithm is the best to tackle our problem. LR obtain as good result as NB but only when trained using the room profile. KNN and SVM do not outperform NB in any scenario evaluated. We performed a statistical analysis to measure the significance of NB classification accuracy and F-measure values. Using a one-sided non-parametric Wilcoxon paired test, we confirm the hypothesis that NB has greater classification accuracy, and F-measure values, than KNN and SVM ($p = 0.03$).

The NB assumes that all attributes of the examples are independent of each other. Albeit this assumption is false in most existing tasks, NB often performs classification satisfactorily (D’SOUZA; ANSARI, 2018; XU, 2018). NB classifiers require a small number of data points to be trained and can deal with high-dimensional data points, favoring this model to our problem (MAHDAVINEJAD et al., 2018). McCallum et al. (1998) explain that in binary cases like ours, the classification estimation is just a function of the sign of the function estimation. Therefore, even when the function approximation is poor, the classification accuracy can be high.

The user profile contains reviews and labels indicating whether the review is describing a good or bad hotel based on the aspect rating. A classifier is trained using the profile, then the reviews in the profile are used as input to evaluate the classifications generated by the classifier. The kappa coefficient measures the agreement between the classification generated by the classifier with the ideal classification. Kappa coefficient is defined by Equation 6.9, where p_o is the observed classification, and p_e the expected one. The expected classification for each review is represented by labels that are associated with them in the profile. Figure 6.7(a) depicts a graph where axis y corresponds to the kappa coefficient values and x represents the classifiers.

$$Kappa = \frac{p_o - p_e}{1 - p_e} \quad (6.9)$$

The higher the kappa value is, the better is the agreement between the observed and expected classifications. As shown in Figure 6.7(a), the classifiers trained using NB present the best results. In particular, classifiers trained using the service, and cleanliness profile obtain the best kappa values. Meanwhile, classifiers trained with the SVM algorithm achieve an intermediate result between the ones trained with NB and KNN. Moreover, classifiers trained using DT obtained negative kappa values as depicted in Figure 6.7(b), when using service and room profiles. According to Vanbelle (2016), a negative kappa indicates that the agreement between the observed and expected classifications is

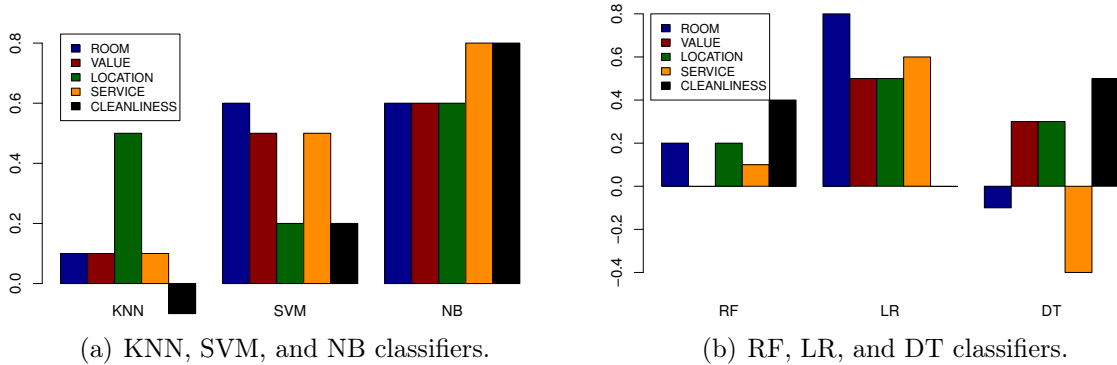


Figure 6.7: Measuring the agreement between the results generated by the classifiers with the expected ones.

worse than expected.

6.2.7 Results

To understand the ranking quality of the personalized query results, we compare the NDCG values obtained when using random keywords and frequent keywords. The Personalized SKPQ (P-SKPQ) is compared with the SKPQ-LD (the same query but without the personalization). The SKPQ-LD employs cosine similarity to measure the text similarity between query keywords and textual description of features. For this reason, we use traditional textual similarity methods such as Jaro-Winkler distance (JW) and Fuzzy score in comparison to the P-SKPQ as well (ASTRAIN; MENDÍVIL; GARITAGOITIA, 2006; COHEN; RAVIKUMAR; FIENBERG, 2003).

Jaro-Winkler is an edit distance between two strings, while the Fuzzy score is a matching algorithm similar to the algorithms implemented in editors such as Sublime Text and TextMate. The Fuzzy score algorithm gives one point for every character matched, while the subsequent matches yield two bonus points. Under these circumstances, a higher score indicates higher similarity.

Figure 6.8 exhibits the arithmetic mean of NDCG@k ($k=5, 10, 15, 20$). For each k value, each query is executed 10 times using different keywords. The arithmetic mean values are reported on the vertical axis. In addition, the personalized query results are obtained using a classifier trained with the “service” profile. The ideal rank results are defined by the hotels with high service aspect ratings in the OpinRank dataset since “service” is the profile chosen to process the query.

It is important to emphasize that SKPQ-LD employs cosine similarity to identify the features description that matches the query keywords. JW and Fuzzy described in Figure 6.8 are the same queries as SKPQ-LD but both replace the cosine similarity with Jaro-Winkler distance, and Fuzzy score to identify the string matching.

The SKPQ-LD presents a poor performance using random query keywords (Figure 6.8(b)). Comparing the SKPQ-LD with the Fuzzy and JW query variations, it is possible

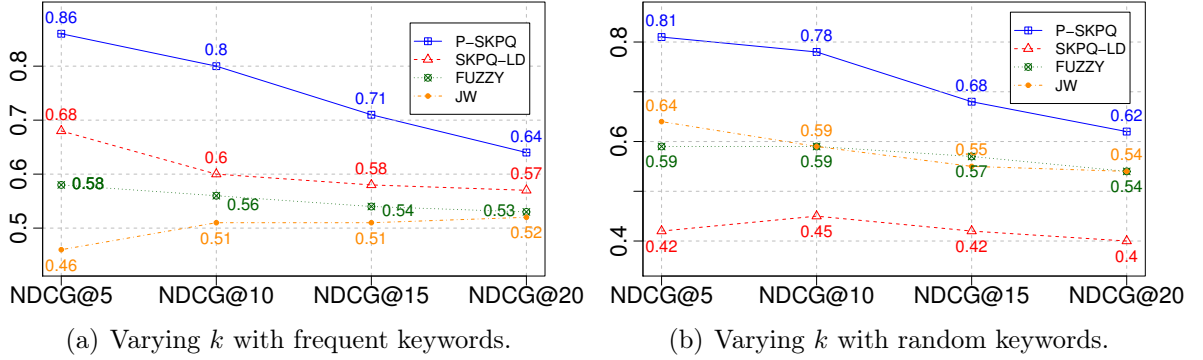


Figure 6.8: Results obtained by P-SKPQ compared to SKPQ-LD, Fuzzy, and JW using Dubai dataset.

to observe that cosine similarity does not provide good results when applying random query keywords. In this case, the cosine similarity compares words used to describe the feature with the query keywords. For example, the words “dog” and “hotdog” have no textual similarity when compared to each other using cosine similarity. In an opposite direction, JW and Fuzzy analyze each character of each word, identifying more similar words as a result. For this reason, JW and Fuzzy identify similarities between “dog” and “hotdog”. Since the random keywords usually have few occurrences in features’ description, it is coherent that JW and Fuzzy obtain better results than SKPQ-LD. This is possible because JW and Fuzzy can identify more features’ description that relates to the query keywords. Despite the use of cosine similarity, P-SKPQ presents the best NDCG values in each evaluated scenario ($k=5,10,15, 20$). These results demonstrate that the personalization improves the query considerably.

We can apply the Equation 6.10 to demonstrate the relative NDCG improvement (as described in Song et al. (2016)) of the proposed approach e_{pro} over the SKPQ-LD e_{other} . In this way, the relative NDCG improvement reaches 92% when comparing P-SKPQ with SKPQ-LD using random keywords, and 33% when using frequent keywords.

$$(e_{pro} - e_{other})/e_{other} \times 100 \quad (6.10)$$

Conversely, SKPQ-LD (cosine similarity) does not perform worse than JW, and Fuzzy using frequent keywords (Figure 6.8(a)). As frequent keywords occur with frequency in the description of features, SKPQ-LD identifies more features with textual relevance to the query keywords. Consequently, SKPQ-LD generates a better query result than JW and Fuzzy. Despite, our proposal (P-SKPQ) still performs better than SKPQ-LD in each evaluated scenario. Equally important, we observe that P-SKPQ obtains lower NDCG values as the k value increases in both experiments illustrated in Figures 6.8(a) and 6.8(b). This outcome is expected since the number of objects in the query result hinders the results ordering process. The more objects are in the query result, the more difficult it is to select the best object for the first rank position. Since there are more objects to assess, the system has more chances to misjudge the best object for the user.

Figure 6.9 exhibits the average NDCG values obtained by the queries when processed using the London dataset provided by Almeida and Rocha-Junior (2015). The results obtained in Figure 6.9(a) illustrates a new scenario using a bigger dataset. The result achieved by the SKPQ-LD using frequent keywords remain higher than the ones obtained using JW and Fuzzy, as observed in Figure 6.8(a). Re-ordering the SKPQ-LD rank still increases the rank quality in all scenarios ($k = 5, 10, 15, 20$), increasing the NDCG values obtained by the P-SKPQ up to 0.11 - a relative NDCG improvement of 13%.

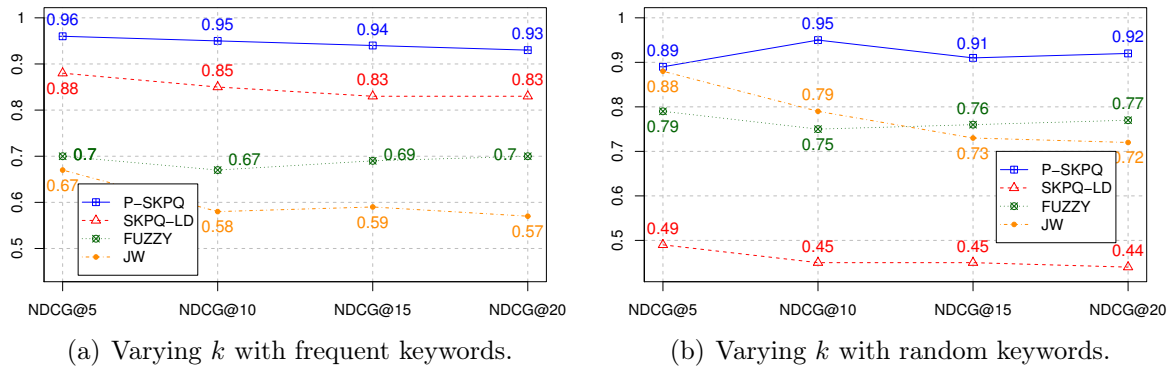


Figure 6.9: Results obtained by P-SKPQ compared to SKPQ-LD, Fuzzy, and JW using the London dataset provided by Almeida and Rocha-Junior (2015).

It is possible to observe that the SKPQ-LD results depicted in Figure 6.9(b) are similar to the ones described in Figure 6.8(b). SKPQ-LD still performs poorly when using random query keywords. However, the P-SKPQ is successful in personalizing and improving the rank quality. JW and Fuzzy perform better in London dataset than in Dubai, but their ranks still do not have the same quality as the ones generated by P-SKPQ. The P-SKPQ rank quality is observed when the lowest NDCG value achieved by P-SKPQ (0.89) is greater than the highest NDCG value achieved by JW (0.88), and Fuzzy (0.79).

6.2.8 Discussion

Personalization concerns the construction of user profiles aiming at providing personalized services. The proposal employs a query results personalization in order to present the best object to the user in the first position of the rank. The results described in Section 6.2 indicates that the personalization improves the query results significantly. Comparing SKPQ-LD with P-SKPQ, it is possible to observe that in every scenario evaluated, the P-SKPQ performs better than the other queries.

The result obtained indicates that there is sufficient information in a user profile to personalize the query. The approach requires a reasonable number of reviews (twenty) in the user profile to personalize the query, providing a meaningful query result improvement. However, P-SKPQ suffers from the cold start problem because the algorithm depends on the user profile (LIKA; KOLOMVATSOS; HADJIEFTHYMIADES, 2014).

Our evaluation considers a user who is active in the system and has submitted a number of reviews. This way, the P-SKPQ may obtain inferior results when the user profile is empty or containing fewer reviews than the number employed in the experimental evaluation.

In essence, module 2 achieves the SO 3 of this thesis by personalizing the SKPQ-LD using textual classifiers. It presents a strategy to re-order the POIs, contributing to answering RQ 1. The results validate the use of a user profile to model the user preference (RQ 3) in combination with module 1 to improve the order of items in the rank generated by the query (RQ 4).

6.2.9 Limitations and Points of improvements

Despite the promising results, our approach has some limitations. It is possible to observe that P-SKPQ suffers from the cold start problem (LIKA; KOLOMVATSOS; HADJIEFTHYMIANES, 2014) because our evaluation considers a user who is active in the system and has submitted several reviews. In this way, the P-SKPQ may achieve low NDCG values when the user profile is empty or containing fewer reviews than the number employed in the experimental evaluation.

Many solutions applied in Recommender Systems can be adapted to our scenario (PENG et al., 2018) to mitigate the cold start problem. For instance, Li et al. (2017) apply context data obtained from social networks to create an initial user profile. Similarly, reviews generated by similar users can be used to create the initial user profile. Besides, it is possible to interview new users about their interest (ZHANG et al., 2015) or analyze the user demographic information (AL-SHAMRI, 2016).

It is still important to consider that users preferences are dynamic. In fact, the user preference changes over time, but the P-SKPQ is not sensitive to this changes. For example, a user interested in a cheap hotel during his/her youth will likely look for comfortable or family hotels once he/she has kids. Thus, it is worth to analyze old reviews in the user profile differently from the new ones. In this fashion, (ZENG et al., 2018) propose a temporal user profile model that considers that the user's interest changes on time intervals. To achieve this goal, they associate each topic⁹ of every user with a continuous distribution. Then, they construct a temporal user profile to predict the relation between users and items.

We can state that the P-SKPQ depends considerably on the user profile. In order to enable the SKPQ-LD to generate results that satisfy the user preference, the profile creating process is crucial. In our study, this profile is created by a specialist who selects the reviews from real users following a proper methodology. However, in a real system, the user profile can contain reviews that do not describe the user preference for many reasons such as typos and even emotional distress. For example, a user can submit a bad review of a hotel because he had argued with other costumers. In case the system does not offer an option to delete such reviews, the user will not be described accurately.

⁹Topic refers to abstract "topics" that occur in a collection of documents.

Table 6.6: Experiment setting. Default values in bold.

Parameters	Values
Number of results (k)	5, 10 , 15, 20
Number of keywords	1 , 2, 3, 4, 5
α	0.05, 0.25, 0.5 , 0.75, 0.95
Cities	Berlin, London, New York, Los Angeles

6.3 MODULE 3 EVALUATION: PROBABILITY-BASED RANKING FUNCTION

A POI average rating represents its average reputation to the user (LEI; QIAN; ZHAO, 2016). Applications employ user ratings to evaluate a rank (CARVALHO; CALADO; CARVALHO, 2017; GANESAN; ZHAI, 2011; LIU et al., 2011; SARWAT et al., 2014). Notably, Ganesan and Zhai (2011) demonstrate that average numerical ratings given by web users are a good approximation to human judgments. Thereby, we consider the POI rating as the POI’s relevance to the user. In addition to satisfy the query constraints (radius and textual relevance), we want to evaluate the query rank order and the query rank accuracy. Therefore, the NDCG measures the average user satisfaction based on the POI order in the rank, while the Tau coefficient is adopted to evaluate the overall ranking accuracy. The following sections describe the results achieved by each algorithm while changing k and the number of keywords.

6.3.1 Experiment Setup

This section presents the methodology applied to process and evaluate the query. In a sequence, the performance of our algorithms and the baselines are evaluated. Table 6.6 presents the experiment settings according to similar settings in the literature (CUI et al., 2019; ANDRADE; ROCHA-JUNIOR, 2019; LEE; LEE; HWANG, 2017; XIE et al., 2016). The values in bold are default values.

The queries are processed using 20 unique keywords for each dataset. Half of the keywords are frequent, and the other half is random. The frequent keywords are selected from a set containing words along with their occurrence in POIs’ descriptions (similar to an inverted file (ZOBEL; MOFFAT, 2006)), while the random keywords are randomly selected from the textual representation of features. The terms “aquarium” and “phone” are examples of random keywords used in this experiment. The query radius is set at 1 km in all queries processed. The code and all requirements to execute and evaluate the module are available at <<https://github.com/JoaoAlmeida/Enhancing-SKPQ>>.

6.3.1.1 Baseline Methods The algorithms that exploit the probabilistic rank function are denoted by PSM (Probability-Based Search Model) and PRR (Probability-Based Ranking Re-Order). We compare them with SKPQ-LD and INF. The SKPQ-LD utilizes a ranking function composed only by the textual relevance between the query keywords and the features’ description. The maximum score of a feature in the vicinity of the

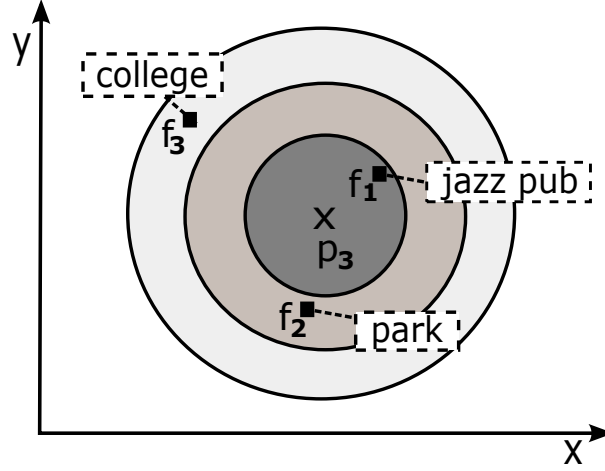


Figure 6.10: Example of POI neighborhood considering the influence score.

POI defines the POI's score (see Equation 5.1). In summary, SKPQ-LD is the same as PSM, without considering the user visiting probability Pr' (see Section 2.2). We compare the algorithms with SKPQ-LD to understand the impact of employing the user visiting probability Pr' .

Influence score is a traditional rank score (ALMEIDA; ROCHA-JUNIOR, 2016; ROCHA-JUNIOR et al., 2010; YIU et al., 2007; YIU et al., 2011) that also considers the POI distance to a feature. Since the user visiting probability is related to the distance between POI and features, the influence score is suitable for comparison. Hence, we create the INF algorithm - an algorithm similar to PSM that uses $\tau_{inf}(p, q)$ (Equation 6.11) instead of $\tau^P(f, q)$ to define the score of a POI. We compare INF with PSM and PRR to measure the proposed ranking function performance against a traditional one.

Under INF, the score of a POI $\tau_{inf}(p, q)$ is defined as the maximum feature influence score in the set of features nearby the POI. The influence score penalizes features far from the POI multiplying $2^{-dist(p,f)/q.r}$ with the textual relevance of the feature to the query keywords $\theta(f.d, q.d)$; where $dist(p, f)$ is the distance of a POI p to a feature f , and $q.r$ is the query radius. This ranking function creates regions of influence, described in Figure 6.10 by the circles around the POI p_3 . The more is the distance between the feature and the POI, the less is the influence. The query radius controls how rapidly the influence score decreases with distance. In contrast, PSM and PRR employ the visiting probability Pr' to decrease the score of POIs distant from its relevant feature (as described in Section 5.4.2).

$$\tau_{inf}(p, q) = \max\{\theta(f.d, q.d) \cdot 2^{-dist(p,f)/q.r}\} \quad (6.11)$$

6.3.2 Datasets

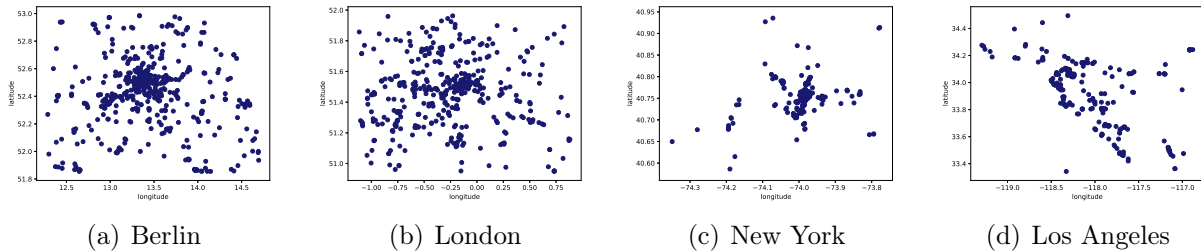
Four datasets are employed to process the query: Berlin (BE), London (LDN), Los Angeles (LA), and New York (NY). Each dataset contains a set of points of interest (1) and features (2). Moreover, the Google Maps data (3) is accessed through the Google Places

Table 6.7: Datasets characteristics.

Datasets	POIs	Features
Berlin	777	619916
London	672	463067
Los Angeles	257	528674
New York	244	214113

API to evaluate the query results. The methodology to obtain and process the datasets is detailed in a sequence:

1. **Points of Interest:** OpenStreetMap (OSM) is a collaborative map of the world available under an open license. Mapzen is an online service that prunes OSM data at the region of individual cities. We use this service to obtain updated data from four different cities: Berlin, London, New York City, and Los Angeles. For each city, we process the data to select only spatial objects whose area is not relevant, excluding roads, rivers, and other objects that are not relevant to the experiment. The set of Points of Interest P is composed of spatial objects whose category in OSM is “hotel”. Table 6.7 describes the number of POIs in each dataset, while Figure 6.11 illustrates their spatial distribution.

**Figure 6.11:** POIs distribution in real datasets.

2. **Features:** We use Linked Data to increase the details in the description of features, concatenating data from different linked sources (Algorithm described in Section 5.2). Thereby, SKPQ-LD searches in LinkedGeodata for features that are in the spatial vicinity of each POI. The query radius defines the vicinity area around the POI where occurs the search for features. The textual score θ is computed only for the features inside that area. Table 6.7 describes the number of features in each dataset.
3. **Ground-truth:** Considering that a POI average rating represents its average reputation to the user (LEI; QIAN; ZHAO, 2016), we analyze the relevance of a rank to the user by considering the average rating of each POI in the rank. Thereby, the ideal rank is the one whose POI with the highest average rating in Google Maps is

also at the top of the query-generated rank. The Normalized Discounted Cumulative Gain (NDCG) and the Tau coefficient are calculated using the user ratings of the POIs in the query rank to indicate the ideal position of each POI.

6.3.3 Metrics

Two classic metrics are used to assess the query rank: Normalized Discounted Cumulative Gain (NDCG) and Kendall’s Tau Coefficient. The NDCG is particularly suitable for search applications since it accounts for multilevel relevance, indicating whether the query rank orders the POIs correctly. The NDCG is described in Section 6.1.3.

Kendall’s Tau Coefficient (TAU) measures the ranking accuracy (LIU et al., 2017; ZHANG et al., 2018) when we consider the POI rating. On that premise, we test all possible pairs of POIs in a query rank P_k . For a target query rank, we consider the score (y_i), and rate (r_i) of POI (p_i) pairs. The POI pair (p_i, p'_i) is considered *concordant* if both $y_i > y'_i$ and $r_i > r'_i$, or if both $y_i < y'_i$ and $r_i < r'_i$. Otherwise, if both $y_i > y'_i$ and $r_i < r'_i$, or if both $y_i < y'_i$ and $r_i > r'_i$ the POI pair (p_i, p'_i) is considered *discordant*. Since it is possible to occur ties between the POIs ratings and score, we adopt the Tau-b version of Kendall’s Tau (KENDALL, 1945):

$$Tau = \frac{c - d}{\sqrt{(c + d + s_ties) \cdot (c + d + r_ties)}}$$

, where c and d are the number of concordant and discordant POI pairs respectively. s_ties is the number of ties that occur in the rank comparing the POI pair using the score (y_i), and r_ties is the same but using the rate (r_i). We compare the average Tau and NDCG of ranks produced by different ranking functions.

6.3.4 Setting the Alpha Value

First, we execute queries varying alpha to understand which measure (textual relevance θ or visiting probability Pr') in the ranking function would fit better our data to increase the rank accuracy. Alpha values are within the range $[0, 1]$, therefore the multiplication $\alpha \cdot \theta$ (Equation 5.2) indicates the textual relevance weight in the ranking function. For example, $\alpha = 0.25$ means that the textual relevance weight in the ranking function is 25% while the other 75% corresponds to the visiting probability. Figure 6.12 reports the ranking performance in terms of NDCG and Tau achieved by the proposed algorithms while varying the alpha values considering the default parameters described in Table 6.6.

We hypothesize if considering the θ and Pr' equally in the ranking function would provide the best rank results. Comparing the balanced ranking function ($alpha = 50\%$) with other alpha values results in a small variation on the NDCG achieved by both algorithm proposals (< 0.03). In particular, PRR obtained similar performance in terms of TAU and NDCG with different alpha values. PRR re-order the query result, thereby the alpha parameter only affects the POIs in the rank instead of all POIs in the search space. We observe that the POIs rarely change their position in the rank because of changes in the alpha value. For this reason, changing alpha has little effect in the POI rank

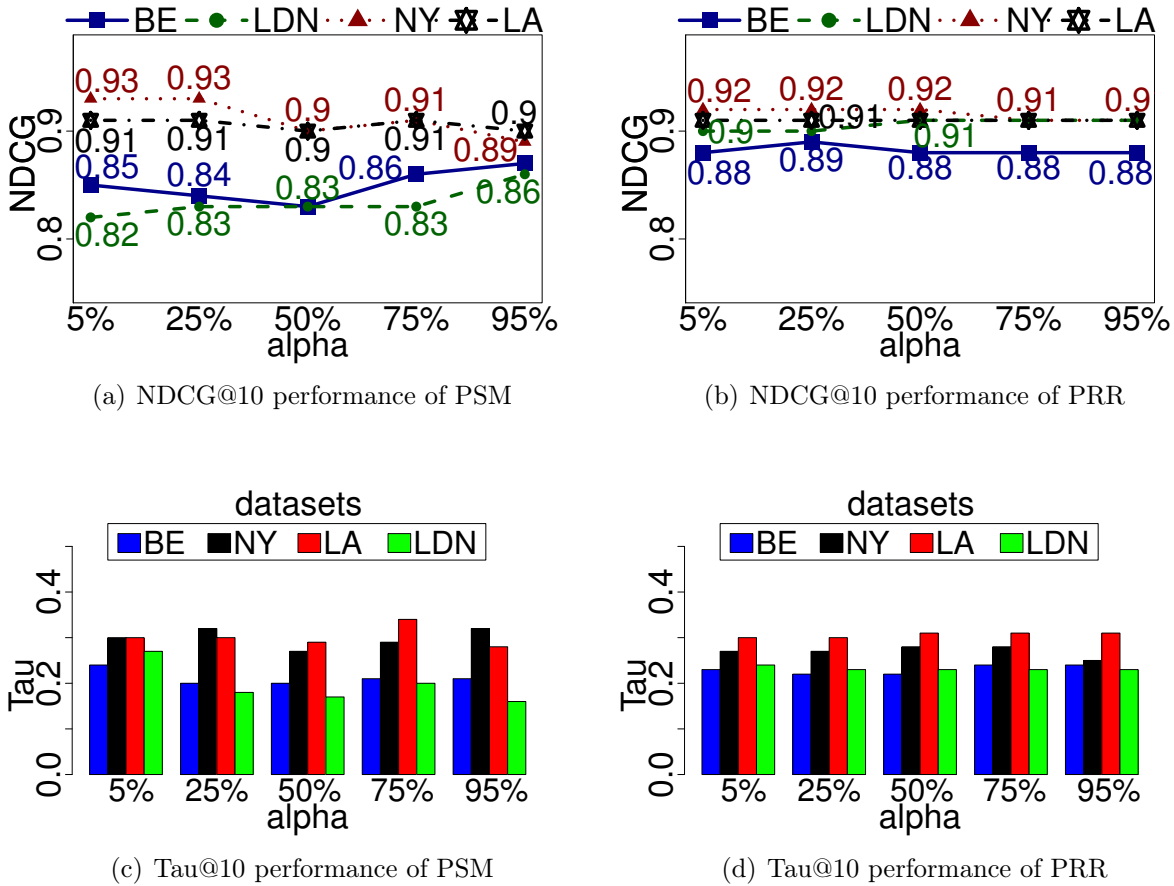


Figure 6.12: NDCG and Tau values achieved by PSM and PRR w.r.t different α values and different datasets.

generated by PRR, resulting in ranks with similar NDCG and TAU values as described in Figures 6.12(b) and 6.12(d). Considering the ranking performance of PRR and to not bias the result to θ or Pr' , we set $\alpha = 50\%$ as default in the following experiments.

6.3.5 Experiment 1: Average User Satisfaction

Figures 6.13 and 6.14 summarize the set of experiments to evaluate the query rank considering different methods under different rank sizes. Each figure depicts the average NDCG and Tau values achieved by PSM, PRR, INF, and SKPQ-LD algorithms. We observe in Figure 6.13 that both proposed algorithms achieve larger or equal NDCG values than the baselines (INF and SKPQ-LD), except for BE when $k = 5$. Considering this query parameter, the NDCG value of PSM is outperformed by SKPQ-LD ($P=0.02$), while the difference between PRR and SKPQ-LD is not statistically relevant ($P=0.24$).

We notice in Figure 6.13 that the NDCG value decreases as the rank size (k) increases. This is expected because as the number of POIs in the rank increases, it also augments the difficulty to order them correctly. All analyzed algorithms decrease their NDCG values

while the rank size increases. The main change in the NDCG achieved by SKPQ-LD occurs in LA, where it decreases from 0.95 ($k = 5$) to 0.88 ($k = 20$), a decrease of 7.37% caused by the rank size increase. Likewise, the worst change in the NDCG value of PSM is a decrease of 7.45% in LA, while PRR achieves -8.60% in LDN.

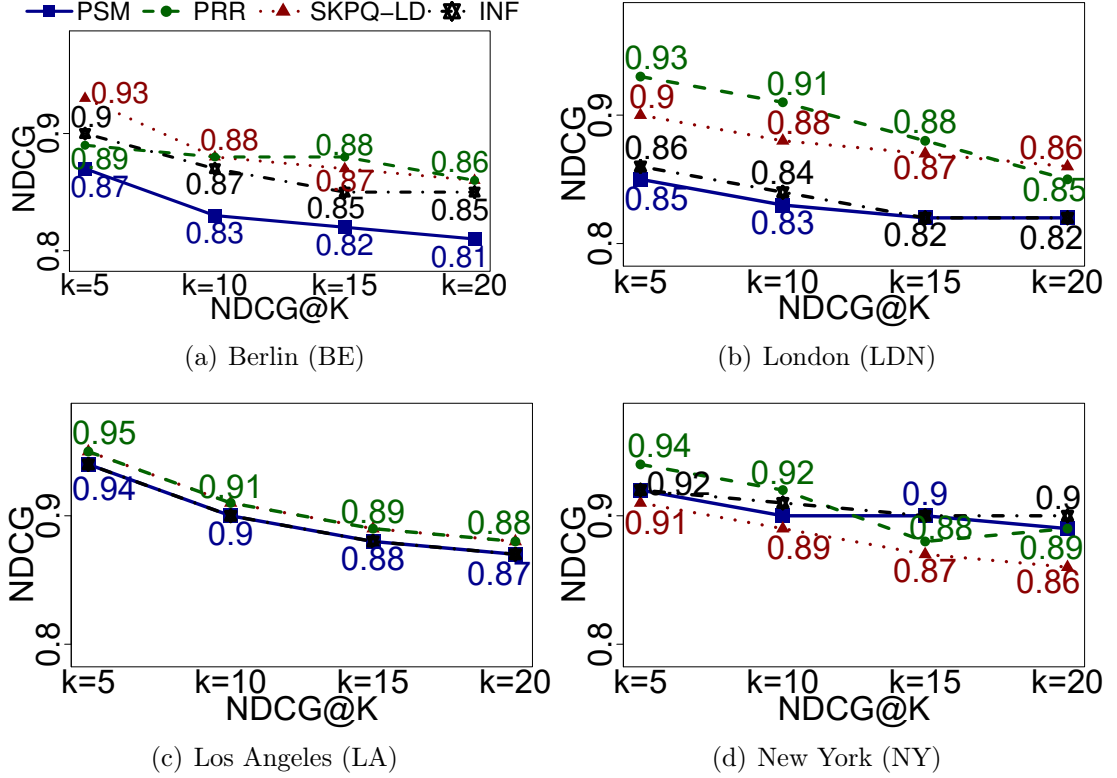


Figure 6.13: NDCG varying the rank size k and the datasets.

Measuring the ranking performance with Tau (Figure 6.14), we observe that PRR achieves larger or equal TAU values than the baselines in every query configuration evaluated, except NY when $k \geq 15$ ($P=.04$; $P=.02$). The worst performance of PRR in comparison to SKPQ-LD is -36% ($k = 20$ in NY), while PSM achieves -8% in NY ($k = 20$). In comparison with INF, PRR and PSM achieve their worst performance in LA at $k = 5$, resulting in -31.25% and -20.83% percentage change ($P=.004$; $P=.003$). SKPQ-LD only outperforms the proposed algorithms in three of sixteen query parameters configuration, while INF accomplishes the same in four of sixteen evaluated scenarios.

In contrast, PSM and PRR greatly improve the rank accuracy in other evaluated scenarios. PSM achieves a Tau value of 0.5 in NY ($k = 5$) while SKPQ-LD achieves 0.18. Alternatively, PRR achieves 0.29 in BE ($k = 5$), while SKPQ-LD obtains 0.07. Therefore, PSM improves SKPQ-LD by 177.78% ($P=.006$) in NY ($k = 5$), and PRR improves it by 314.29% ($P=.003$) in BE ($k = 5$). Although both proposed algorithms do not improve the rank accuracy in all rank sizes evaluated, we observe they can improve the performance more than worsen it, achieving a satisfactory trade-off. On average, PRR improves the NDCG values of SKPQ-LD by 0.93%, while PSM decreases it by 2.67%.

Nevertheless, PSM and PRR considerably improve the average Tau values by 69.55% and 68.49%, respectively.

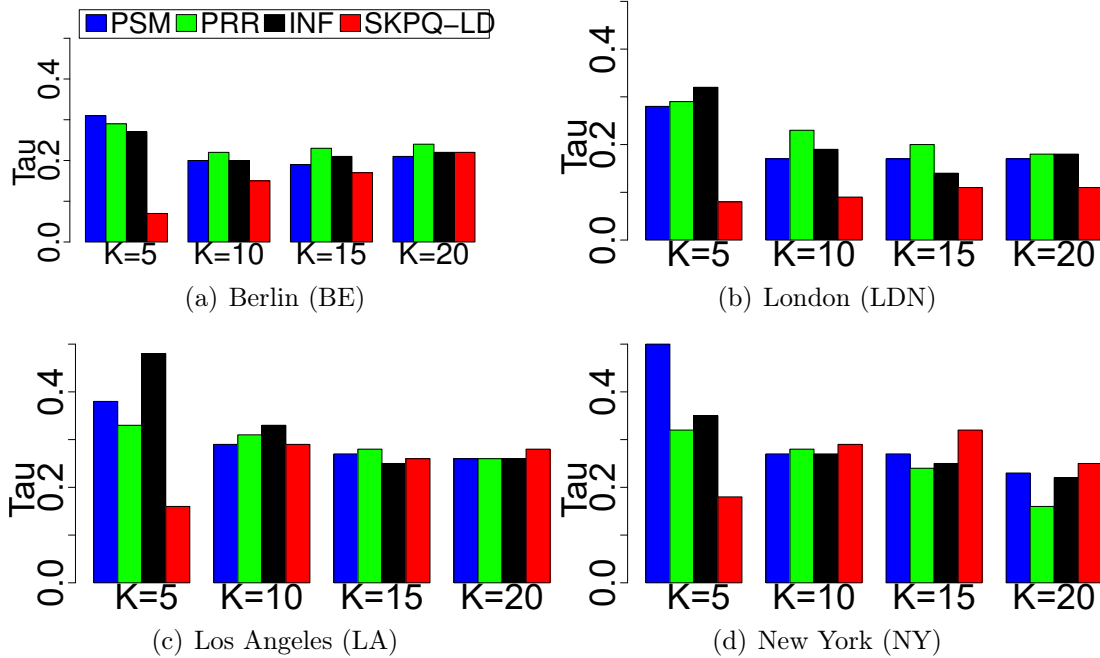


Figure 6.14: Tau varying the rank size k and the datasets.

The Tau performance improves when the number of ties in the rank positions decreases and the POIs are correctly ordered considering the POI relevance to the user. The SKPQ-LD obtains ties in its rank because it considers only the textual relevance in the ranking function (see Section 5.4.2). For this reason, it often defines the same score for different POIs. Since many POIs are nearby to each other (as seen in Figure 6.11), they share similar neighborhoods. In this scenario, employing only the textual relevance is not enough to differentiate one POI from another, resulting in a high number of ties in the query rank, leading to a poor Tau performance. The problem is usually alleviated as the size of the rank increases because the number of POIs with no intersection in their neighborhoods increases, resulting in fewer ties. Additionally, the SKPQ-LD achieves its best Tau values in LA and NY, where there are fewer POIs spatially close to each other than LDN and BE, as observed in Figure 6.11. This behavior hardly occurs in PSM, PRR, or INF that consider the spatial constraint to define the score.

6.3.6 Experiment 2: Varying the Number of Keywords

Keyword queries usually contain a small number of keywords (JANSEN et al., 1998). According to Liao and Tajima (2019), most queries submitted to web search engines include only one or two terms. Similarly, Kacprzak et al. (2019) argue that single term queries represent almost half of the queries collected from the official UK government Open Data portal and the Office for National Statistics of the UK. Considering the

number of keywords in these reports, Experiment 2 evaluates PSM and PRR performance while varying the number of query keywords. The experiment consists of executing queries containing 2 to 5 keywords, using different ranking functions to evaluate their NDCG and Tau performance.

Figures 6.15 and 6.16 depict the NDCG and Tau values obtained while varying the keyword number. All evaluated methods have similar NDCG performance while varying the number of keywords. Their NDCG values vary within the range of 0.01 to 0.06. However, the comparison between the NDCG performances of the algorithms is not statistically significant ($P > .05$). In contrast to Experiment 1, the trend here is to increase the NDCG performance when the number of keywords increases instead of decrease the performance when the rank size increases. By increasing the number of keywords, usually improves the description of the information need, resulting in a better NDCG performance.

Although not considering the spatial constraint, SKPQ-LD improves its Tau performance as the number of keywords increases in the majority of the scenarios presented in Figure 6.16, especially using the Berlin dataset. Despite this improvement caused by the number of keywords, PSM and PRR achieve better or similar Tau values than SKPQ-LD. PRR improves the TAU performance of SKPQ-LD by 287.5% in London with four keywords ($P < .001$), and PSM improves it by 187.5% ($P = .008$) with the same query parameters.

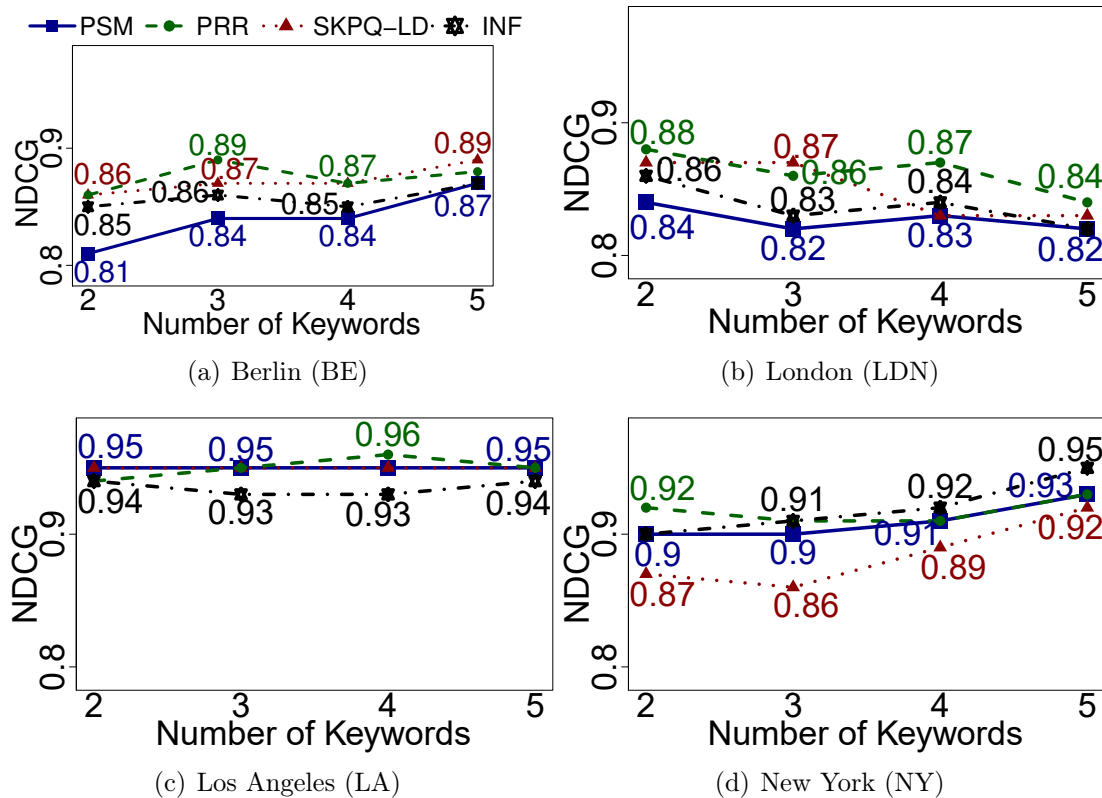


Figure 6.15: NDCG varying the number of keywords.

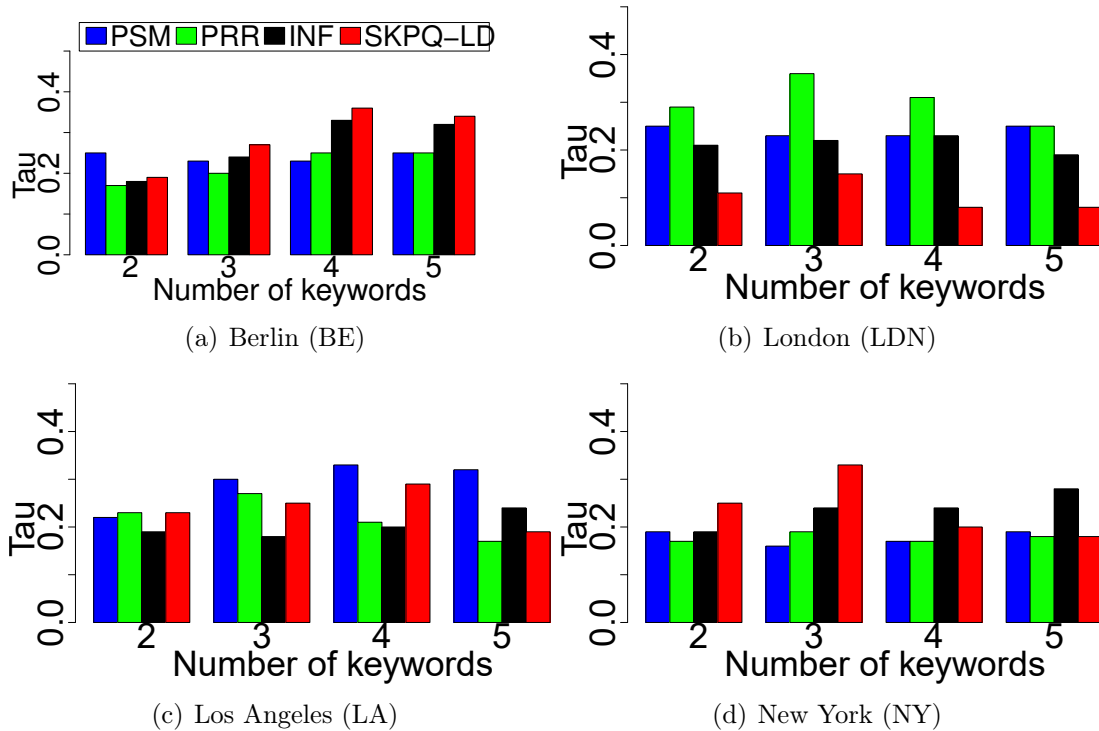


Figure 6.16: Tau varying the number of keywords.

In Section 6.3.5, we discuss that SKPQ-LD was unable to differentiate the feature in the intersection of spatial neighborhoods of POIs. Now, we observe that using more keywords to describe the feature attenuates the problem and improves the Tau and NDCG performances of SKPQ-LD. However, this improvement does not contribute to SKPQ-LD surpass the performance of PRR, demonstrating the effectiveness of PRR as the number of query keywords increases. By varying the number of keywords, PRR improves the NDCG performance of SKPQ-LD by 1.16% on average, while PSM decreases the performance by 0.94%. Moreover, PRR improves the Tau performance of SKPQ-LD by 36.91%, and PSM improves it by 24.87%. Comparing with the traditional function score (INF), PRR improves its average Tau performance by 3.15%, while PSM decreases it by -0.53% . Since INF does not consider only the textual relevance like SKPQ-LD, it is able to generate better ranks than the latter. Despite this improvement, PRR still achieves better Tau ranking performance than INF.

6.3.7 Discussion

The POIs' spatial neighborhood is a key factor in PRR and PSM algorithms. Datasets that contain a considerable amount of POIs with intersecting neighborhoods benefit more PRR than PSM, while PRR still maintains a competitive performance in other datasets. The POIs distribution in Figure 6.11 suggests the spatial areas where PSM can improve the query rank. Considering the scenarios in Experiments 1 and 2, PRR achieves better

(or similar) performance than the other approaches in this context (22 out of 32 query configurations evaluated with NDCG).

The PRR demonstrates that employing the Pareto probability in the rank re-ordering can improve the SKPQ-LD rank. Comparing the PRR with a similar method employed in the literature (INF) culminates in a better ranking order (NDCG) and a better ranking accuracy (Tau). As to Experiments 1 and 2, PRR improves the NDCG value of INF by 1.87%, and the Tau performance by 2.41%, on average.

Applying the Pareto distribution to model the user visiting probability is a simple and effective method. It does not require training a user model (XIA; GONG; ZHU, 2011), saving the system from the training computational costs. Additionally, there is no need to store user preferences in a matrix such as some recommender systems (LIAN et al., 2014). We also observe that PSM and PRR improve the SKPQ-LD, PRR outperforms the INF algorithm, and it is statically better to satisfy an average user than the baselines.

In summary, the evaluation of module 3 validates the algorithm employed to achieve the SO 4. It presents another strategy to search and re-order the rank generated by the query, contributing to answering the RQ 1. Module 3 also employs the description enhancement employed in module 1, presenting another answer to RQ 4.

6.3.8 Limitations and Points of Improvements

The POIs' spatial neighborhood is a key factor in PRR and PSM algorithms. Datasets that contain a considerable amount of POIs with intersecting neighborhoods benefit more PRR than PSM, while PRR still maintains a competitive performance in other datasets. The POIs distribution in Figure 6.11 suggests the spatial areas where PSM can improve the query rank. Considering the scenarios in Experiments 1 and 2, PRR achieves better (or similar) performance than the other approaches in this context (22 out of 32 query configurations evaluated with NDCG).

The PRR demonstrates that employing the Pareto probability in the rank re-ordering can improve the SKPQ-LD rank. Comparing the PRR with a similar method employed in the literature (INF) culminates in a better ranking order (NDCG) and a better ranking accuracy (Tau). As to Experiments 1 and 2, PRR improves the NDCG value of INF by 1.87%, and the Tau performance by 2.41%, on average.

Applying the Pareto distribution to model the user visiting probability is a simple and effective method. It does not require training a user model (XIA; GONG; ZHU, 2011), saving the system from the training computational costs. Additionally, there is no need to store user preferences in a matrix such as some recommender systems (LIAN et al., 2014). We also observe that PSM and PRR improve the SKPQ-LD, PRR outperforms the INF algorithm, and it is statically better to satisfy an average user than the baselines.

The Pareto distribution, as applied in this thesis, represents an average user spatial preference. We do not estimate the distribution parameters to better fit a specific user movement data like Zhu et al. (2015). Hence, we do not provide personalized information retrieval when adopting PSM or PRR. Lately, users are more aware of the need for privacy in their data; for this reason, many proposals suggest solutions to anonymize personal data of the user (JADALLAH; AGHBARI, 2019; KIM; KIM; CHANG, 2019; ZHANG et

al., 2019; ZHU; LIU; LI, 2017). In fact, our algorithms can improve the SKPQ-LD rank without using personal data, avoiding privacy concerns.

6.4 THE COVID-19 GEO-MONITOR USE CASE

The CIDACS/Bahia is part of a governmental institute that conducts research and development of multidisciplinary products within the areas of epidemiology, statistics, bioinformatics, and computing. One current goal of CIDACS/Bahia is to monitor and provide information about the spread of the COVID-19 in Brazil. In collaboration with an epidemiologist and two computer scientists who work on CIDACS/Bahia, we developed a use case to aid authorities in monitoring the number of COVID-19 infections in specific neighborhoods of the city. Moreover, the use case indicates the nearest Basic Health Unit (UBS) from the patient’s residence or location. Textual enhancement (as described in Section 5.2) improves the description of places in the neighborhood of COVID-19 patients when this information is available. Considering these functions, the use case can help authorities to plan actions against the COVID-19 outbreak, intensifying oversight, or enforcing movement restrictions in specific areas of the city; also, it can help to manage health supplies to the UBS that has more cases in the neighborhood. In a sequence, we describe the use case named as COVID-19 GEO-MONITOR in detail.

6.4.1 Ranking POI considering patients with COVID-19 in their neighborhood

The use case executes a SKPQ-LD adaptation to search for POIs in which neighborhoods have the most cases of COVID-19. To adapt the SKPQ-LD, we consider the patients’ location as a feature to the POI instead of places associated with a textual description. Thereby, the most relevant POI to this adapted query is that with the highest number of infected patients in the neighborhood. Since we do not have access to real patient data, we simulate the patient dataset from CIDACS/Bahia as the features dataset. The POIs dataset P is composed of places in a city (e.g. places in São Paulo). The user can choose between search considering all POIs in the city, or just a category of POIs inside of P (e.g. universities or beaches). The search considering only a category of POIs enables the user to narrow the search space. Algorithm 5 describes the query executed in the use case to search for POIs containing infected patients in the neighborhood.

For each POI $p \in P$, the S2I returns a set of tuples containing information about the patients in the neighborhood of p (line 6). The algorithm counts the number of patients returned by the S2I using the iterator (lines 7-9). In the use case, we consider the number of patients as the score of p because we want the POI with the most number of cases in its neighborhood in the first position of the rank (line 10). Therefore, $\tau(p, q) = |\text{patients}|$. Lines 11-16 are similar to Algorithm 1.

The use case is a Web application where the user can submit the query parameters and visualize the query results. If the user wants to restrict the search to a specific category of POI, he/she defines it using the drop-down menu described in Figure 6.17. Then, he/she has to define the query radius, that describes the neighborhood area of each POI. Otherwise, he/she defines only the query radius to generate a rank that includes all

Algorithm 5: Processing SKPQ-LD in the use case COVID-19 GEO-MONITOR.

Input: $q = (q.r)$
Output: Heap H that maintains the k best points of interest.

```

1  $H \leftarrow \emptyset$ 
2  $q.k \leftarrow 20$ 
3 for each  $p \in P$  do
4    $\tau(p, q) \leftarrow 0$ 
5    $patients \leftarrow 0$ 
6    $iterator \leftarrow s2i.findPatients(p).iterator()$ 
7   while  $iterator.hasNext()$  do
8      $patients ++$ 
9   end
10   $\tau(p, q) \leftarrow patients$ 
11  if  $|H| < q.k$  OR  $\tau(p, q) > H.peekMin().score$  then
12     $H.add(p)$ 
13    if  $|H| > q.k$  then
14       $H.removeMin()$ 
15    end
16  end
17 end
18 return  $H$ 

```

POIs in the city.

The results are presented to the user in two ways: the POIs' neighborhood are printed on a map (Figure 6.18(a)), and then they are also printed in a table (Figure 6.18(b)). After the query processing, each POI in the query result, and its respective neighborhood, are printed on a map. The neighborhoods printed in Figure 6.18(a) highlights the area in the city where exists more cases; enabling specialists to visualize risk areas inside the city. The results printed in the table (Figure 6.18(b)) lists the name of the POIs and the number of people infected in the neighborhood of these POIs. Each row contains two buttons to further present detail about the POI in that row. In the use case, the result size $q.k$ (see Section 5.4.2) of SKPQ-LD is 20 to simplify the interface in order to all POIs appear in the results table at once.

Figure 6.19 describes the POI detail after the user clicks on the magnifier button. A new window appears (Figure 6.19(a)) illustrating the POI location (red mark) and its respective neighborhood (circle in red). If the user clicks or moves the mouse over the red mark, a dialog window presents the POI description extracted from DBpedia (Figure 6.19(b)). The description extracted from DBpedia extends the description presented in the table early to the user (Figure 6.18(b)), increasing the detail in the POI description.

Another window appears if the user clicks on the plus button (Figure 6.20), presenting a list with all patients in the neighborhood of the POI and their respective distance from the POI. The list can be ordered considering the distance of the user to the POI by

Display a rank of the selected place, considering confirmed cases of the disease in the neighborhood

Select the place

Define the search area:
Ex: 1000

Figure 6.17: Query parameters to search for places with the most number of COVID-19 infections in its neighborhood.

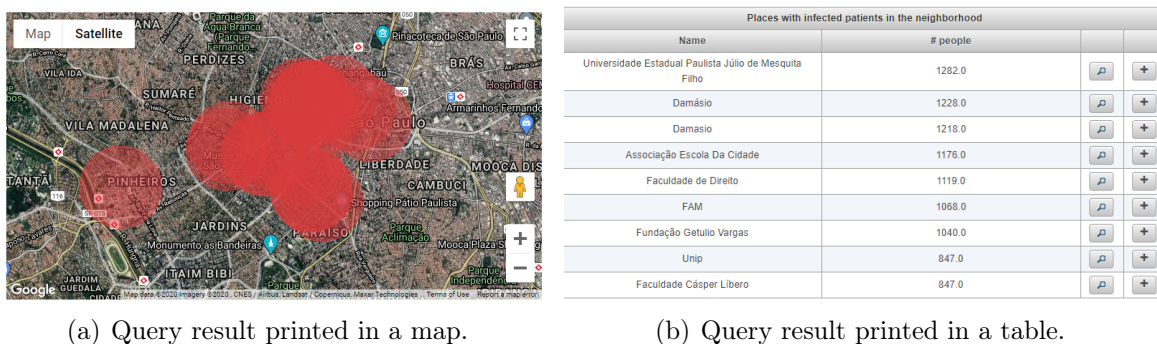


Figure 6.18: Query result presented in a map and a table. The first column of the table is the name of each POI, and the second is the number of infected persons in the neighborhood. The magnifier button leads to a window to visualize the POI in detail (Figure 6.19), while the plus sign button leads to a list containing the patients' names and distances to the POI (Figure 6.20).

clicking on the “Distance” column name.

6.4.2 Presenting the nearest UBS to a patient

The CIDACS/Bahia identifies the need to associate a patient location to a Basic Health Unit (UBS), also informing specialists information about the UBS such as the number of Intense Care Units or the number of drugs available in that Health Unit. Adopting the well-known nearest-neighbor algorithm (YIU et al., 2007; LE et al., 2019), the application presents on a map the location of the nearest UBS to the patient. The user can submit the location (e.g. latitude and longitude) of the patient by clicking on the map or typing it directly in the input box. The application access a database containing information about all UBS in the city. Then, the Euclidean distance from the patient to each UBS is calculated. In the end, the UBS with the shortest distance is highlighted on the map.

Figure 6.21 illustrates the application's output after a patient location is submitted. The red mark on the map represents the user location, while the medicine symbol mark represents the UBS location. The UBS's description appears in a dialog box after a click

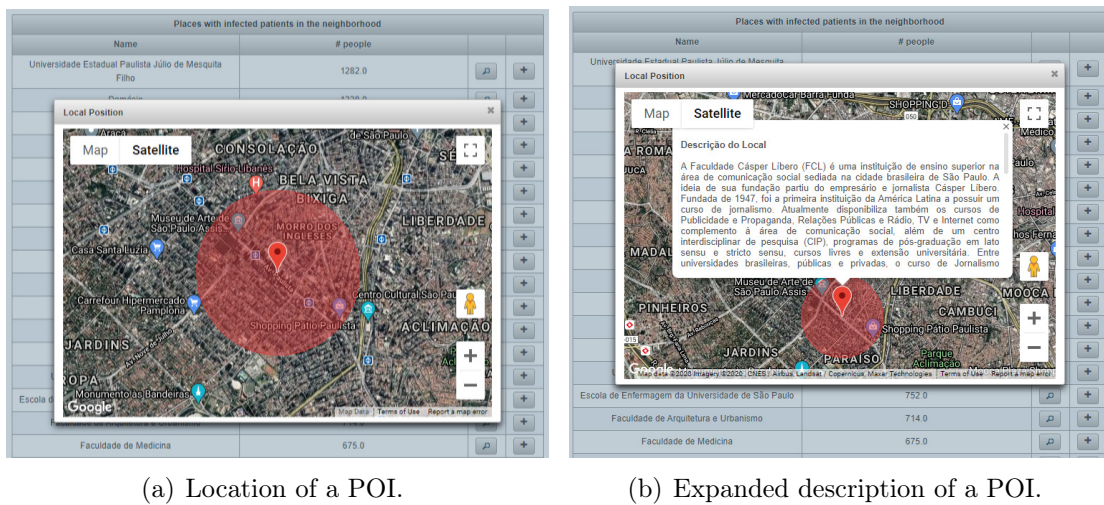


Figure 6.19: Detailed view of a specific POI in the query result. The LOD description of the POI appears after a click on the red mark. The circle in red represents the POI's neighborhood.

on the medicine mark. This description is enhanced by our proposal accessing DBpedia when the information is available. Moreover, the dialog box presents technical data about the UBS, such as the number of medical equipment and medicines available in that unit.

6.5 EVALUATION OF THE COVID-19 GEO-MONITOR USE CASE

A multidisciplinary team of specialists is responsible for the use case project: three Ph.D. in Computer Science, one in Epidemiology, and the author of this thesis who developed the prototype. Initially, several meetings were conducted by this team to define the scope of the prototype and how the proposals described in this thesis could contribute to the work in CIDACS/Bahia. CIDACS/Bahia is a governmental organization hosted in Salvador - Bahia that provides information about city health to the government. Since the outbreak of the COVID-19 pandemic, the organization monitors health unities resources (e.g. medical equipment, medicines) and the spread of the disease.

The scope of the use case defines two tasks for the application. First, ranking places according to COVID-19 cases in their neighborhoods. Second, associate a patient location to the nearest UBS from his/her location. Currently, CIDACS/Bahia identifies patients infected with COVID-19 executing SQL queries on a database. This way, it is possible to identify the location of an infected patient; however, it is not possible to discover the places around the infected patient using only the organization's database. The application order places according to the number of COVID-19 cases in their neighborhood and also enhance the textual description of these places. Regarding the second task, CIDACS/Bahia does not have any software to associate a patient to the nearest UBS, neither enhance its description nor present technical data about it.

Places with infected patients in the neighborhood	
Name	# people
Universidade Estadual Paulista Júlio de Mesquita	
Filho	
Damásio	
Damasio	
Associação Escola Da C	
Faculdade de Direito	
FAM	
Fundação Getulio Varg	
Unip	
Faculdade Cásper Líbe	
Centro Universitário Maria A	
Centro Universitário Maria A	
PUC Ciências Exatas	
Fundação Getulio Varg	
Universidade Presbiteriana M	

Infected patients in the neighborhood	
Name	Distance
Patient 508	0.0028699999999979298
Patient 509	0.0
Patient 685	2.9689999999860106E-4
Patient 686	0.0
Patient 687	0.0
Patient 689	0.0
Patient 690	0.0
Patient 831	2.916999999982295E-4
Patient 832	0.00240069999999034
Patient 835	0.0
Patient 836	0.0
Patient 837	0.0
Patient 842	0.0018047499999980232
Patient 843	0.004253458637683354
Patient 884	0.0
Patient 988	0.00816899999999876

Figure 6.20: List of patients in the POI's neighborhood and its distance to each patient.

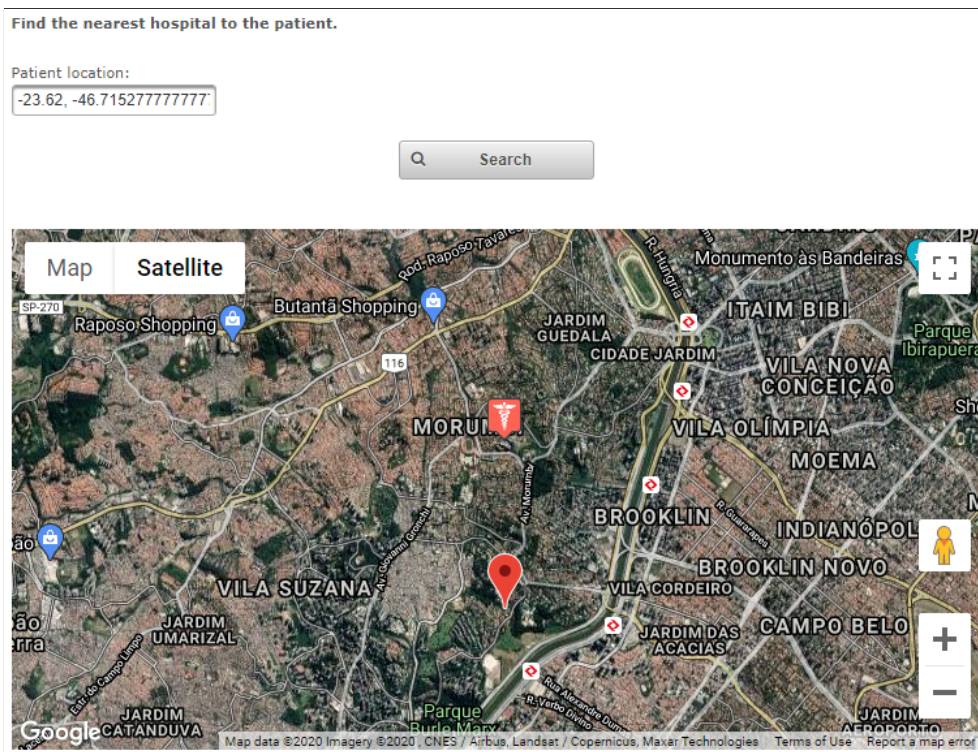


Figure 6.21: Overview of our approach to automatically improve query results.

6.5.1 Experiment Setup

To evaluate the application resulted from the use case, we presented the prototype to five specialists from CIDACS/Bahia. Three did not get involved in the development process of the prototype, while the other two did. These two specialists contributed with a theoretical background about the organization data model and epidemiology concepts, needs, and limitations. We submitted a survey to collect the opinion of these five specialists about the prototype functionalities. This survey has no intention to be statistically representative because of the small number of responses, but it effectively collects the opinion of specialists who work on the topic. In summary, five persons watched the application’s presentation, but one did not answer the survey. Table 6.8 describes the role of each specialist in the use case’s development and the survey.

Table 6.8: Participants characteristics.

Participant	Answered the survey	Involved in the project	Field of work
<i>Person</i> ₁	yes	yes	Address geocoding and data linkage
<i>Person</i> ₂	yes	yes	Epidemiology researcher
<i>Person</i> ₃	yes	no	Address geocoding and data visualization
<i>Person</i> ₄	yes	no	Address geocoding
<i>Person</i> ₅	no	no	Epidemiology researcher

To collect the specialists’ opinions, we ask them to answer a survey using a 5-point linear scale and writing a small text in sequence to describe any suggestion or limitation they have identified. Considering the small number of participants, we do not calculate Cronbach’s alpha to measure the survey reliability. This metric requires at least fifty participants to investigate the psychometric properties of the survey (HENRI; MORRELL; SCOTT, 2018). This number of participants in the survey is not possible to achieve, considering only the staff of CIDACS/Bahia.

6.5.2 Datasets

The patients’ data stored in CIDACS/Bahia databases are protected by law (General Law on Protection of Personal Data - law No. 13.709/2018), so it only can be accessed and manipulated inside the organization’s infrastructure, inside an offline environment, to guarantee the privacy of patients’ data. The specialists who collaborate with us in the use case provided knowledge about how data is collected, stored, and manipulated. By knowing the information we could access in the organization’s dataset, we simulate the organization’s dataset to develop our application.

A SQL database stores the data about UBS and patients. Each instance in the

UBS table contains the location (latitude and longitude) and the UBS's name, while the patients' table stores his/her name, location, address, date of infection, CPF (taxpayer number), and RG (general registry). CPF and RG are common identifications numbers to citizens in Brazil.

The application accesses the SQL database and stores the patients' data in a S2I. This index enables the application to search for the patient data efficiently, improving the query processing time (see Section 2.8.1). The data about places in the city are collected from OpenStreetMap and stored in a R-tree before the query execution (see Section 2.5.2).

6.5.3 Metric

Linear scales provide a range of responses to a statement. In this thesis, we use the number one to consider an application function useless and the number five to consider it useful. Figure 6.22 illustrates an example of a linear scale used in the survey submitted to the specialists. A full copy of the survey is attached in Appendix C, followed by the raw responses provided by the participants in Appendix D.

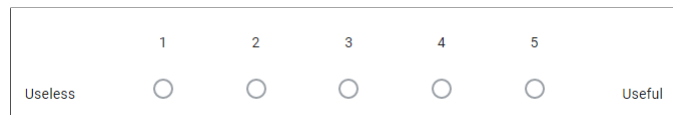


Figure 6.22: Example of linear scale.

6.5.4 Results

Four participants answered the survey: two who participate in the development team and two who only view the presentation about the final version of the prototype. They rated each prototype function and our proposal using a 5-point linear scale that considers five as useful and one as useless. Additionally, the participants answered how they could improve the prototype if they have the opportunity. To conclude the survey, we ask each participant to describe his/her feeling about the prototype presentation understanding. A copy of the survey is attached in the Appendix C.

Figure 6.23 describes the specialists response to the prototype functions. We asked the participants about the usability of the functions; and if they could solve problems encountered by specialists of CIDACS/Bahia. The participants submitted their responses using a scale of one to five, indicating the relevance of the function to them. Two out of four participants consider the rank of places a useful tool to deal with the new coronavirus pandemic (Figure 6.23(a)). Regarding the patient association to the nearest UBS, three out of four considered the tool useful (Figure 6.23(b)). No participant considered any function useless nor irrelevant.

We also ask the specialists to identify any limitations or describe any improvement suggestions for each use case function. Regarding the patient association to the nearest UBS, one participant considers it relevant to identify the user location automatically

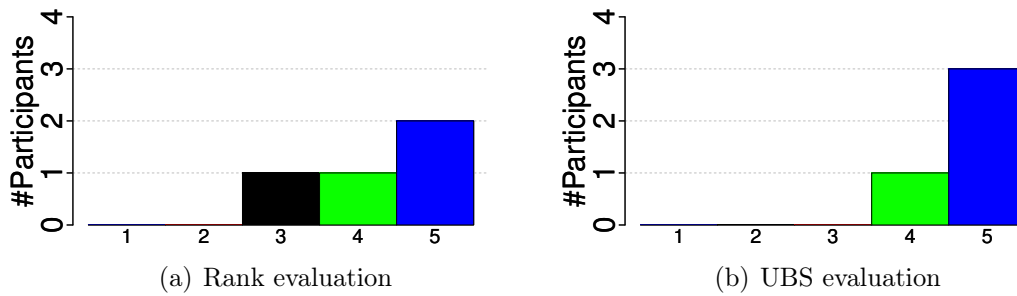


Figure 6.23: 5-point linear scale response of specialists about the use case functions.

instead of entering it manually in the text box or mark the position in the map (Figure 6.21). Another participant rises the concern about the availability of spatial data. The other participants do not identify any limitations nor describe any suggestions.

Concerning the rank of places, one participant suggests we include a query that generates a rank of places also considering the incidence rate, instead of only the number of positive cases of COVID-19. The incidence rate describes the number of cases during a specific amount of time. Another participant suggests we include the possibility to search for multiple types of places (e.g. bars, hospitals, gyms) instead of select one option in the list (Figure 6.17). Similarly to the other function, a participant also raises the concern about the availability of spatial data when asked about the limitation of the rank of places. One participant does not identify any limitation nor describe a suggestion.

We inform the participants that a relevant contribution of our research in the prototype is the textual description enhancement of POIs. We also explain that, without a proper description, some places can not be considered in the epidemiology analysis about places with COVID-19 spread in their neighborhood. For example, a school could not be identified as a school by a search system; because no label describes the place. All participants in the survey defined this enhancement as useful (Figure 6.24).

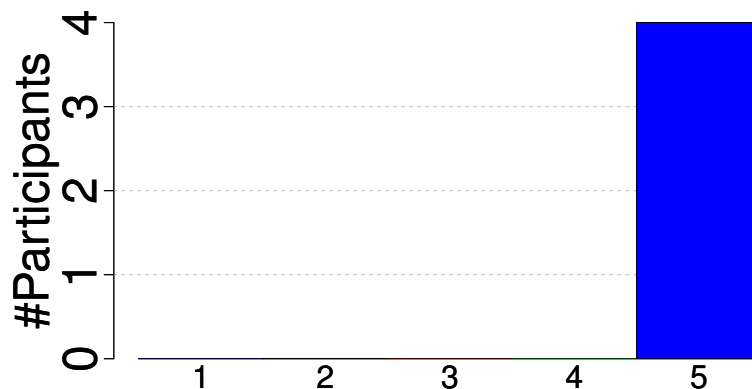


Figure 6.24: Response of specialists about the textual description enhancement of POIs.

Afterward the prototype presentation, we asked the participants to describe their feeling about the prototype. Three out of four are satisfied with the results obtained by

the tools presented, while the other participant defines herself as excited about the use case results, as illustrated by Figure 6.25.

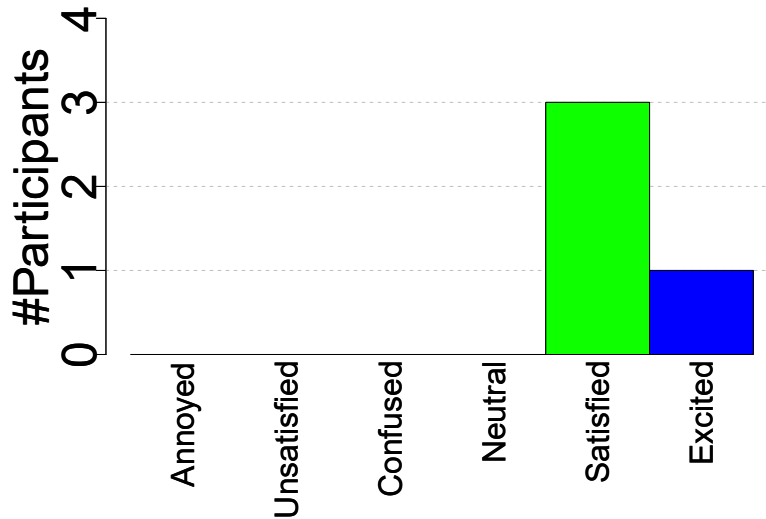


Figure 6.25: Response of specialists about their expectation about the prototype.

6.5.5 Discussion

Considering the specialists' responses in the survey and the requirements described in the use case scope, we understand that the use case can provide useful information about the COVID-19 outbreak inside a city. In this section, we quote each feedback provided by the specialists and discuss them in sequence. The first feedback statement is the following about the association of a patient to the nearest UBS:

It is relevant to identify the user location automatically instead of enter it manually in the text box or mark the position in the map.

Indeed it is a relevant function to add to the application. During the scope definition of the application, we idealize that the patient association to the nearest UBS would be conducted by a UBS's employee, using the application. For this reason, we did not consider automatic identification of the patient's location.

There is lack of spatial data associated to the patients.

This statement was repeated by the same specialist when describing the limitation of both functionalities. Unfortunately, collect data about every citizen with quality is a challenge. We are aware that many patients are registered without addresses and other demographic information such as gender or age. This problem occurs because the patient decides not to provide the information or because it was not collected during patient admission. Many researchers put effort to mitigate this problem and collect accurate information about patients. However, this is not the scope of our thesis. Nowadays, there is an effort in CIDACS/Bahia to geocoding the address of patients. Our application is tailored to access this geocoded information.

Include a query that generates a rank of places also considering the incidence rate, instead of only the number of positive cases of COVID-19.

This statement is a relevant contribution we collected from the survey. Even consulting specialists to develop the application, we did not consider the incidence rate in our analysis. However, to calculate the incidence rate, we need more information about the patients in addition to his/her location. According to Dicker et al. (2011), the incidence rate is a division where the numerator is the number of new cases identified during the observation period, and the denominator is the sum of the time each person was observed. In addition to more complex information, we also need to change the S2I to a spatio-temporal index to store the patients' data, such as GeoSOT (QIAN et al., 2019). It considers time and space in the storage process to enable faster query processing.

Include the possibility to search for multiple types of places (e.g. bars, hospitals, gyms), instead of select one option in the list.

The application already has two options for search: considering all places in the city or selecting one option. Adding one more option to select multiple types of places to search is relevant, and does not demand much effort. Considering multiple choices, the user can control the search space with more freedom.

In essence, the COVID-19 Geo-monitor use case contributes to answering the RQ 2 by exploiting LOD to improve the description of places with infected patients in their neighborhood. Also, it achieves the SO 5 by applying the SKPQ-LD to aid specialists in monitoring the spread of the disease in a particular city.

6.5.6 Limitations and Points of improvements

A major limitation of the application is the availability of spatial data associated with the patients, as mentioned by one of the specialists. Many patients are registered without an address, therefore, these patients are not considered by the application. Moreover, the data can not be manipulated outside the organization's infrastructure (digitally or physically). This limitation hardens the address' geocoding process because it is not possible to use APIs like Google Places that converts an address into spatial coordinates (latitude and longitude) with precision. Therefore, there are a team of specialists in CIDACS/Bahia working to geocode the patients' address inside the organization.

Nowadays, CIDACS/Bahia can convert an address into an area code that represents the patient's neighborhood. The area code does not represent the patient's precise location but describes the neighborhood where his residence is located. The query employed in the use case uses latitude and longitude to describe the patients' location but it can use the area code either.

There are points of improvement described by the specialists in Section 6.5.5 that must be considered, such as consider the incidence rate in the search, enable the choice of multiple types of places, and identify the user's location automatically. Moreover, the query that generates a rank of places considering COVID-19 cases in the neighborhood can be applied to search for other diseases transmitted by viruses, like Dengue and Zika

viruses. These diseases frequently occur in Brazil, for this reason, there are extensive literature and patients data available.

6.6 SUMMARY

This chapter described the datasets employed in the query processing as well as the metrics employed in the query evaluation. It discusses the results obtained in the experimental evaluation, addressing the challenges to improve spatial keyword preference query results. It presents a textual description enhancement based on LOD (SKPQ-LD) that improves SKPQ by 20% when using random query keywords. Then, the personalization algorithm re-orders the preliminary results considering the user preference described in his/her past reviews. The assessment indicates a relative NDCG improvement of the P-SKPQ over the SKPQ-LD of 92% when using random query keywords and 33% when using frequent keywords. We propose another option to re-order the query results considering a probability-based rank function. It achieves an average NDCG performance of 1.04% and Tau performance of 52.70% in comparison with the SKPQ-LD in real-world datasets. The results and strategies discussed in this chapter contribute to answer the research questions RQ 4 and RQ 5. The next chapter concludes the thesis and outlines the main contributions and future work.

PART V

FINAL REMARKS

FINAL REMARKS

The top- k spatial query is a significant class of query that returns a set of POIs in a potentially vast data space. This query class filters the retrieved data in an ordered set of answers, known as rank. There is a significant effort to determine the correct item position in a rank considering user's satisfaction. The item position in the rank must correlate with the user satisfaction regarding that item. Thereby, the item that best satisfies the user must be in the first position of the rank. Usually, it is not trivial to identify the POI that best fulfills the user's need. This thesis described solutions to improve the POI order in the rank generated by top- k Spatial Preference queries. We exploit LOD to expand POI's description, textual classifiers to re-order the query results, and a probabilistic function to describe the user's implicit preference. We showed the efficiency of our approaches through experiments employing real datasets. Next, we present the contributions, impressions, and directions to future work.

7.1 CONTRIBUTIONS

This D.Sc. thesis contains the following contributions:

- *Textual enhancement accessing different datasets (Chapter 5, Section 5.2)*. We proposed and implemented an algorithm to automatically enhance the description of POIs by integrating different Linked Open Data datasets. The algorithm enables the query to find POIs that have small or nonexistent textual descriptions, contributing to the quality of the results generated by the query.
- *Spatial top- k keyword query results personalization (Chapter 5, Section 5.3)*. To further improve the query, we exploited personalization techniques to reorder the query results considering the user's personal and implicit preference. We built user profiles based on user reviews on TripAdvisor to train a textual classifier. We propose and implement an algorithm that employs the classifier to identify if an unknown POI has reviews similar to ones the user had visited. The query results are personalized by reordering the POIs' position, considering the user's implicit preference.

- *Exploiting a probabilistic function to model the user preference (Chapter 5, Section 5.4)*. Considering to describe the average user preference for POIs close to each other, we defined a probabilistic-based rank function. We analyzed the Pareto distribution with real-world datasets and employed it in our novel rank function. We also proposed and developed two algorithms (PSM and PRR) to explore the rank function and process the top- k spatial keyword preference query. PSM explores the search space employing the novel rank function to define the score of a POI. In contrast, PRR adopts the probabilistic-based rank function to reorder the query results generated by the query. This way, the POI position in the rank can reflect the user's implicit preference.
- *Spatial top- k keyword query evaluations (Chapter 6)*. We evaluated all our proposed algorithms by applying them in different contexts and scenarios, considering empiric query evaluation methods, and providing information for further replications. We exploit user ratings from real users and widely adopted metrics to evaluate the query results. Our framework has methods to automatic extract an average user rating to a POI from different data sources such as TripAdvisor and Google Maps. It also automatically evaluate the query results considering the metrics discussed in this thesis.
- *Use case application (Chapter 5, Section 6.4)*. We gathered several specialists from CIDACS/Bahia to develop a use case to monitor the COVID-19 outbreak by processing the SKPQ. We developed a Web application with two functionalities: rank places considering COVID-19 infected patients in their neighborhoods and associate a patient's location to the nearest UBS. We submitted specialists to a survey and collected insightful feedback about the Web application and the query processing.

7.2 IMPRESSIONS

SKPQ is a top- k spatial search that uses keywords to describe the user's interest. Allied to this, the search considers the neighborhood of the POI instead of the POI itself. These search characteristics bring challenges to its processing and evaluation of results because of the number of items and their constraints to be considered in the search process. Thereby, considering the state-of-the-art on this search class, this thesis brings contributions to the field that have not yet been experimented with. To the best of our knowledge, no one had used LOD to increase the textual description of POIs, nor had they personalized top- k spatial keyword queries results. Exploring solutions commonly applied in Recommendation Systems, we also were able to model the user's preference in a new ranking function, using the Pareto distribution.

Tackling the challenge imposed by the COVID-19 pandemic, we teamed up with researchers from government organizations (university and government research center) to develop solutions to monitor the virus spread and aid health authorities in resource management. This use case has potential in the view of the technical and social contributions that emanate from it. Besides, experts say that the use case can be expanded to monitor other endemic diseases in Brazil, such as Zika and Dengue.

The first module proposed for the framework described in this thesis enhances the descriptions of POIs and features using LOD. This module has applicability in other spatial keyword queries and spatial information systems in general instead of just the SKPQ. For example, the first module can be employed by semantic web browsers in order to help users obtain more information about specific POIs. The second module personalizes the query results considering user reviews. It has applicability in spatial information retrieval systems that collect textual feedback from users about items in the system's database, such as Airbnb and AllTrails. Moreover, the third module can improve spatial information systems that do not rely on user feedback. It models the user preference for POIs near to each other, so this novel ranking function can also be applied in a system that plans routes such as OptimoRoute¹ and MyRout Online². By applying the third module, those systems can retrieve POIs considering the distance to other places in the neighborhood. Thereby, a route that satisfies the textual and distance preferences of the user can be planned.

In order to answer the RQ 1, it was investigated three different solutions to re-order the query rank. It is important to notice that the spatial preference queries pose a particular challenge in addition to the traditional personalization process. The user defines two constraints before the query processing: textual relevance and spatial location. Thereby, the system has to satisfy the user's implicit preference and also has to satisfy these two constraints. In contrast to a recommender system, that recommends a movie based on another movie the user has just finished, not requiring additional user explicit interaction such as typing query keywords. Nevertheless, the results achieved by this thesis demonstrate that the combination of textual enhancement with query personalization can improve the order of items in the rank.

The first module answers the RQ 2, presenting an algorithm and SPARQL queries to exploit LOD in order to improve the description of places. This module has potential yet to explore, like consider the semantics of the words during the textual relevance computation. Modules 2 and 3 describe two different techniques to model the user preference, answering RQ 3. Module 2 considers the user's experience to predict POIs he/she may like, while Module 3 models an average user preference. The average preference is described in the literature and is observed by the check-in analysis of a significant number of users (FENG et al., 2017; ZHANG et al., 2018). This way, Module 3 investigates the impact of considering an average preference in the query result.

The personalized query is compared with other queries without personalization. Although they are not personalized, all queries employ the text enhancement described by Module 1. We notice that the combination of personalization with text enhancement achieved better results than the queries using only text enhancement. This way, it is possible to answer RQ 4, considering that it is possible to achieve better rank quality by combining different techniques. To evaluate the modules, we employ user ratings from distinct datasets and real-user query keywords. According to the literature review, the evaluation method applied in the modules represents a good approximation of the user

¹<https://optimoroute.com/>

²<https://www.myrouteonline.com/>

satisfaction (GANESAN; ZHAI, 2011; LEI; QIAN; ZHAO, 2016), answering the RQ 5.

The proposed approaches can have several implications by assisting users in locating POIs that satisfy their needs and in presenting those POIs in a rank. To summarize, this thesis design two novel solutions and experiment with a consolidated technique in a new context applied to spatial keyword preference queries. The results achieved by this thesis can lead researchers and practitioners to novel and motivating solutions. It also provides the framework code coupled with links to all datasets related.

7.3 FUTURE WORK

In this thesis, it was investigated methods to improve the rank consistency of top-k spatial keyword preference queries. However, there is still much to do in this research field. In the following, we describe some future research topics that can motivate further research. Among them, we can mention:

- *Develop a graph to describe different user preferences:* Graph embedding is used in Recommender Systems to jointly learn different user preferences about a POI, integrating geographical, semantic, and social information (ZHANG et al., 2018; ZHU et al., 2015). The hybrid model has demonstrated strong results in the Recommender Systems field. We have evidence to believe that it is possible to use the graph embedding to learn the user preference and process spatial top- k queries, similarly as done by Zhang et al. (2018) that models the user preference to solve the location promotion problem.

In this context, different user preferences can be encoded by embedding methods. For example, Zhang et al. (2018) define a geographical and a preference factor embedding vectors. In order to process the SKPQ, the first step is to identify the POIs that have textual relevance to the query keywords and are inside the POI's neighborhood. Then, the ranking function could consider the embedding vectors to improve rank order instead of considering only the textual relevance.

- *Hybrid index to store spatio-textual linked open data:* Each day more applications that exploit LOD are developed. However, there is no effort to develop hybrid indexes (such as S2I) to store spatio-textual linked data efficiently. Different from S2I, this new index has to store the data considering the tuple pattern intrinsic from Linked Data. A hybrid structure like that can efficiently prune the search space, avoid accessing all POIs in the database. Considering the related studies on hybrid indexes, a new index that considers Linked Data has the potential to reduce the query processing cost in at least one order of magnitude.

Moreover, the LOD is accessed from online repositories in this thesis. An online repository has several limitations, such as the number of queries that can be processed and server availability. The hybrid index can efficiently store data from online repositories in the local disk. Thereby, the hybrid index can serve as a cache to avoid access the same information several times from the online repository. Therefore, the hybrid index can improve the query processing time and can store the data

locally. In this scenario, an update protocol must be defined to update the data stored in the index when changes occur in the online repository.

- *Extend the use case to consider spatio-temporal data:* The specialists on epidemiology have provided in the survey significant insights on how to improve the use case. The Web application can be adapted to consider other diseases than COVID-19 and can use the incidence rate instead of only the number of disease occurrence to search for contamination risk areas. The incidence rate requires identifying the number of cases during a time period to be calculated. For this reason, a new index that considers temporal information has to be incorporated in the application to enable the incidence rate calculation during the query processing. Thereby, a spatio-temporal index enables the system to prune the search space to improve the query processing time because it considers only the POIs that have a high incidence rate and satisfies the spatial criteria.

A temporal SKPQ can return the best spatio-temporal-textual objects ranked in terms of proximity to the query location and proximity to the query time. For example, it can retrieve the top-k POIs with the most infected patients in the neighborhood in January 2021. Therefore, a temporal SKPQ can contribute to the extended use case that considers the incidence rate to rank the POIs. To the best of our knowledge, a temporal SKPQ has never been proposed before.

- *Improve the number of LOD repositories* This thesis employs two LOD repositories, DBpedia and LinkedGeoData. They are used to extend the description of POIs. As future work, we plan to investigate the DBpedia integration with other LOD repositories, improving the description of POIs further. The results achieved by the modules indicate that a more detailed description can improve the retrieval process. Under those circumstances, DBpedia enables SKQP to find more POIs that satisfy the user, and it also improves the POI's order in the rank (according to the POIs' rating). By defining a set of heterogeneous LOD repositories, the system can detail the POI's description further, possibly leading to satisfactory results.

For example, EventKG³ provides information for events and their temporal relations, while YAGO⁴ contains general knowledge about people, cities, countries, movies, and organizations. The combination of these two LOD repositories with DBpedia can lead to a better description for POIs, extending the first module of this thesis. Thereby, the results achieved by the first module can be compared with the one generated by its extension. This results analysis can contribute significantly to the discussion about POIs' description enhancement.

- *Expand the POI representation with semantic information:* Cosine similarity requires an exact keyword match between the POI description and the query keywords to consider the POI textually relevant to the user. This relevance method may lead to few or no results to be retrieved because of the diversified textual

³<<http://eventkg.l3s.uni-hannover.de/>>

⁴<<https://yago-knowledge.org/>>

expressions. To overcome this issue, we presented Module 1 to improve the description of POIs. However, the SKPQ still can not retrieve the POIs whose description is synonym but literally different to those in query keywords, such as “theater” and “cinema”. For this reason, we consider investigating models to understand the semantic meanings of textual descriptions.

For example, Qian et al. (2018) apply the Latent Dirichlet Allocation model to understand the semantic meanings of textual description formed by words related to topics. They apply this probabilistic model in top-k spatial keyword queries and evaluate query time performance and I/O. Different from SKPQ, this type of query only considers the user location. We aim to investigate this topic and also conduct a quality evaluation of the rank considering the average user evaluation.

7.4 DISSEMINATION

The D.Sc. study described in this thesis originated the publications listed below. The content of each publication is primarily described in the Chapters 5 and 6.

Paper 1. Presents the preliminary results of our method to enhance the SKPQ accuracy using Linked Open Data. ALMEIDA, J. P. D. de; DURÃO, F. A. *Improving the Spatial Keyword Preference Query with Linked Open Data. In: Anais Estendidos do XXIV Simpósio Brasileiro de Sistemas Multimídia e Web, 2018. v. 24. p. 19-24.*

Paper 2. Details the approach to enhance the SKPQ and discusses all obtained results in the evaluation. ALMEIDA, J. P. D. de; DURÃO, F. A.; COSTA, A. F. da. *Enhancing Spatial Keyword Preference Query with Linked Open Data. Journal of Universal Computer Science, v. 24, n. 11, p. 1561-1581, 2018.*

Paper 3. From a collaborative work, we exploit Web features for relevance feedback. DURÃO, F. A., ALMEIDA, J. P. D., SANTOS, D., SOUZA, P. R., SCHJØNNING, C., RASMUSSEN, R. (2019). *Exploiting Web Features for Relevance Feedback. In: The Americas Conference on Information Systems - AMCIS, 2019. v. 1.*

Paper 4. Describes and discusses the proposed personalization technique to improve the SKPQ results. ALMEIDA, J. P. D. de; DURÃO, F. A. *Personalizing the Top-k Spatial Keyword Preference Query with Textual Classifiers. Expert Systems with Applications, Elsevier, v. 162c, p.113841, 2020.*

Paper 5 [Submitted, under review]. Detailed report about the probability-based rank function. ALMEIDA, J. P. D. de; DURÃO, F. A. *Exploiting Pareto distribution for user modeling in location-based information retrieval. Expert Systems with Applications, Elsevier, 2021.*

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*. Seattle, USA, 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*. Johor, Malaysia, 2018. p. 282–297.
- AGGARWAL, C. C. *Machine learning for text*. Cham, Switzerland: Springer, 2018.
- AL-SHAMRI, M. Y. H. User profiling approaches for demographic recommender systems. *Knowledge-Based Systems*, Elsevier, v. 100, p. 175–187, 2016.
- 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*. Portland, USA, 2012. v. 46, n. 1, p. 2–32.
- ALMEIDA, J. P. D. de. *Consulta Espacial Preferencial por Palavra-chave*. Dissertation (M.Sc.) — Universidade Estadual de Feira de Santana, Feira de Santana, Brazil, 2015.
- ALMEIDA, J. P. D. de; ROCHA-JUNIOR, J. B. Top-k spatial keyword preference query. *Journal of Information and Data Management*, v. 6, n. 3, p. 162–177, 2016.
- ALMENDROS-JIMÉNEZ, J. M.; BECERRA-TERÓN, A.; TORRES, M. Integrating and querying openstreetmap and linked geo open data. *The Computer Journal*, Oxford University Press, v. 62, n. 3, p. 321–345, 2019.
- ANDRADE, C. M. V. de; ROCHA-JUNIOR, J. B. Popularity-based top-k spatial-keyword preference query. In: *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*. Rio de Janeiro, Brazil: [s.n.], 2019. p. 505–512.
- ANGLES, R.; GUTIERREZ, C. The expressive power of sparql. In: SPRINGER. *International Semantic Web Conference*. Karlsruhe, Germany, 2008. p. 114–129.
- APUKE, O. D. Quantitative research methods: A synopsis approach. *Kuwait Chapter of Arabian Journal of Business and Management Review*, American University, v. 33, n. 5471, p. 1–8, 2017.
- ASTRAIN, J. J.; MENDÍVIL, J. G. de; GARITAGOITIA, J. R. Fuzzy automata with moves compute fuzzy measures between strings. *Fuzzy Sets and Systems*, Elsevier, v. 157, n. 11, p. 1550–1559, 2006.

- ATTIQUE, M.; KHAN, A.; CHUNG, T.-S. *espak: Top-k spatial keyword query processing in directed road networks*. In: *International Conference on Extending Database Technology/Database Theory Workshops*. Venice, Italy: [s.n.], 2017.
- BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern information retrieval: the concepts and technology behind search*. 2th. ed. England: Pearson, 2011.
- 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*. Barcelona, Spain, 2010. p. 119–126.
- BARUFFOLO, A. R-trees for astronomical data indexing. In: *8th Astronomical Data Analysis Software and Systems*. Kona, United States: [s.n.], 1999. v. 172, p. 375.
- BAYER, R.; MCCREIGHT, E. Organization and maintenance of large ordered indexes. In: SPRINGER. *Software pioneers*. Berlin, Germany, 2002. p. 245–262.
- 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 International Conference on Management of Data*. Atlantic City, USA: 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.
- BERNERS-LEE, T. *Design issues: Linked data*. 2006. Available at: <<http://www.w3.org/DesignIssues/LinkedData.html>>. Accessed: 2021-02-26.
- BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. Uniform resource identifier (uri): Generic syntax. *Network Working Group: Fremont, CA, USA*, Internet Engineering Task Force (IETF), 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. In: *Semantic services, interoperability and web applications: emerging concepts*. [S.l.]: IGI global, 2011. p. 205–227.
- BORZSONY, S.; KOSSMANN, D.; STOCKER, K. The skyline operator. In: IEEE. *Proceedings of 17th International Conference on Data Engineering*. Heidelberg, Germany, 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.

BOUIDGHAGHEN, O.; TAMINE, L.; BOUGHANEM, M. Personalizing mobile web search for location sensitive queries. In: IEEE. *12th International Conference on Mobile Data Management*. Luleå, Sweden, 2011. v. 1, p. 110–118.

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)*. Heraklion, Greece: Springer, 2010.

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

CAI, P. et al. Influence constraint based top-k spatial keyword preference query. In: *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*. Sanya, China: ACM, 2019. p. 1–6.

CAO, X. et al. Spatial keyword querying. In: SPRINGER. *International Conference on Conceptual Modeling*. Florence, Italy, 2012. p. 16–29.

CARAMIA, M.; DELL’OLMO, P. *Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level, Sustainability, and Safety with Optimization Algorithms*. Cham, Switzerland: Springer Nature, 2020.

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

CARVALHO, A.; CALADO, P.; CARVALHO, J. P. Combining ratings and item descriptions in recommendation systems using fuzzy fingerprints. In: IEEE. *International Conference on Fuzzy Systems (FUZZ-IEEE)*. Naples, Italy, 2017. p. 1–6.

CHANG, C.-W.; CHEN, C.-D.; CHUANG, K.-T. Queries of k-discriminative paths on road networks. *Knowledge and Information Systems*, Springer, p. 1–30, 2019.

CHEN, L. et al. Spatial keyword query processing: an experimental evaluation. In: VLDB ENDOWMENT. *39th International Conference on Very Large Data Bases*. Trento, Italy, 2013. p. 217–228.

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

CHENG, R.; KALASHNIKOV, D. V.; PRABHAKAR, S. Evaluating probabilistic queries over imprecise data. In: *Proceedings of the International Conference on Management of Data (ACM SIGMOD)*. San Diego, USA: [s.n.], 2003. p. 551–562.

- CHIVADSHETTI, P.; SADAFALÉ, K.; THAKARE, K. Content based video retrieval using integrated feature extraction and personalization of results. In: IEEE. *International Conference on Information Processing (ICIP)*. Pune, India, 2015. p. 170–175.
- CHO, E.; MYERS, S. A.; LESKOVEC, J. Friendship and mobility: user movement in location-based social networks. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. San Diego, USA: ACM, 2011. p. 1082–1090.
- CHOMICKI, J. Preference formulas in relational queries. *ACM Transactions on Database Systems (TODS)*, ACM, v. 28, n. 4, p. 427–466, 2003.
- CHOMICKI, J. *Preference Queries*. New York, USA: Springer, 2018. 2784–2787 p.
- 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.
- CONG, G.; JENSEN, C. S.; WU, D. Efficient retrieval of the top-k most relevant spatial web objects. In: VLDB ENDOWMENT. *35th International Conference on Very Large Data Bases*. Lyon, France, 2009. v. 2, n. 1, p. 337–348.
- CUI, Q. et al. Distance2pre: Personalized spatial preference for next point-of-interest prediction. In: SPRINGER. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Macau, China, 2019. p. 289–301.
- 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. Forbes, 2017. Available at: <<https://bit.ly/3r55FX6>>.
- DICKER, R. C. et al. Principles of epidemiology in public health practice: An introduction to applied epidemiology and biostatistics. US Department of Health and Human Services and others, Atlanta, USA, v. 8, 2011. Available at: <<https://bit.ly/37TI1VU>>.
- DIETZ, J. L. *Enterprise Ontology*. Heidelberg, Berlin: Springer, 2006.
- D’SOUZA, K. J.; ANSARI, Z. Big data science in building medical data classifier using naïve bayes model. In: IEEE. *International Conference on Cloud Computing in Emerging Markets (CCEM)*. Bengaluru, India, 2018. p. 76–80.
- DURAO, F. *Exploiting Tag-Based Personalization for Recommendation on Social Web*. Thesis (Ph.D.) — Aalborg University, Aalborg, Denmark, 2012.
- FENG, S. et al. Poi2vec: Geographical latent representation for predicting future visitors. In: *Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, USA: [s.n.], 2017.

- FERNÁNDEZ-TOBIÁS, 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*. Chicago, USA, 2011. p. 25–32.
- FERRÉ, S. Sparklis: a sparql endpoint explorer for expressive question answering. In: *ISWC posters & demonstrations track*. Trentino, Italy: [s.n.], 2014.
- FIELDING, R. et al. *RFC2616: Hypertext Transfer Protocol–HTTP/1.1*. [S.l.]: RFC Editor, 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. Thesis (Ph.D.) — Universidade de São Paulo, 2019.
- FREUDENBERG, M. *DBpedia version 2016-10*. [S.l.], 2019. Available at: <<https://wiki.dbpedia.org/develop/datasets/dbpedia-version-2016-10>>.
- GANESAN, K.; ZHAI, C. Opinion-based entity ranking. *Information Retrieval*, 2011.
- 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. In: *Proceedings of the International Conference on Management of Data (ACM SIGMOD)*. New York, USA: ACM, 1984. v. 14, n. 2, p. 47–57.
- HAASE, P. et al. A comparison of rdf query languages. In: SPRINGER. *International Semantic Web Conference*. Berlin, Germany, 2004. p. 502–517.
- HAGEN, P.; MANNING, H.; SOUZA, R. Smart personalization. *Forrester Research, Cambridge, MA*, 1999.

HAN, X. et al. Tkap: Efficiently processing top-k query on massive data by adaptive pruning. *Knowledge and Information Systems*, Springer, v. 47, n. 2, p. 301–328, 2016.

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

HARMAN, D. Information retrieval evaluation. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, Morgan & Claypool Publishers, v. 3, n. 2, p. 1–119, 2011.

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*. Basel, Switzerland: [s.n.], 2011.

HENRI, D.; MORRELL, L.; SCOTT, G. Student perceptions of their autonomy at university. *Higher Education*, Springer, v. 75, n. 3, p. 507–516, 2018.

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.

IOANNAKIS, G. et al. Retrieval—an online performance evaluation tool for information retrieval methods. *IEEE Transactions on Multimedia*, IEEE, v. 20, n. 1, p. 119–127, 2017.

JADALLAH, H.; AGHBARI, Z. A. Spatial cloaking for location-based queries in the cloud. *Journal of Ambient Intelligence and Humanized Computing*, Springer, v. 10, n. 9, p. 3339–3347, 2019.

JANSEN, B. J. et al. Real life information retrieval: A study of user queries on the web. In: ACM. *SIGIR Forum*. New York, USA, 1998. v. 32, n. 1, p. 5–17.

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

JIANG, S.; FERREIRA, J.; GONZALEZ, M. C. Activity-based human mobility patterns inferred from mobile phone data: A case study of singapore. *Transactions on Big Data*, IEEE, v. 3, n. 2, p. 208–219, 2017.

JIN, P. et al. Bag-of-embeddings for text classification. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*. New York, USA: [s.n.], 2016. v. 16, p. 2824–2830.

KACPRZAK, E. et al. Characterising dataset search—an analysis of search logs and data requests. *Journal of Web Semantics*, Elsevier, v. 55, p. 37–55, 2019.

- KARAM, R.; MELCHIORI, M. Improving geo-spatial linked data with the wisdom of the crowds. In: ACM. *Proceedings of the joint International Conference on Extending Database Technology/Database Theory workshops*. Genoa, Italy, 2013. p. 68–74.
- KARPATHIOTAKI, M. et al. Prod-trees: semantic search for earth observation products. In: SPRINGER. *European Semantic Web Conference*. Crete, Greece, 2014. p. 374–378.
- KELES, I. *Spatial Keyword Querying: Ranking Evaluation and Efficient Query Processing*. Thesis (Ph.D.) — Aalborg Universitetsforlag, Aalborg, Denmark, 2018.
- KENDALL, M. G. The treatment of ties in ranking problems. *Biometrika*, Oxford University Press, v. 33, n. 3, p. 239–251, 1945.
- KIM, H.-I.; KIM, H.-J.; CHANG, J.-W. A secure knn query processing algorithm using homomorphic encryption on outsourced database. *Data & Knowledge Engineering*, Elsevier, v. 123, p. 101602, 2019.
- KIM, J. G.; HAUSENBLAS, M. 5(star) open data. 2019. Available at: <<https://5stardata.info/en/>>.
- KLYNE, G.; CARROLL, J. J. Resource description framework (rdf): Concepts and abstract syntax. 2006.
- KUZI, S. et al. Analysis of adaptive training for learning to rank in information retrieval. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. Beijing, China: [s.n.], 2019. p. 2325–2328.
- 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*. Brighton, England: [s.n.], 1987. v. 87, p. 1–4.
- LE, S. et al. Balanced nearest neighborhood query in spatial database. In: IEEE. *International Conference on Big Data and Smart Computing (BigComp)*. Kyoto, Japan, 2019. p. 1–4.
- LEE, J.; LEE, D.; HWANG, S.-W. Crowdk: Answering top-k queries with crowdsourcing. *Information Sciences*, Elsevier, v. 399, p. 98–120, 2017.
- LEI, X.; QIAN, X.; ZHAO, G. Rating prediction based on social sentiment from textual reviews. *Transactions on multimedia*, IEEE, v. 18, n. 9, p. 1910–1921, 2016.
- 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*. Macau, China, 2018. p. 350–358.

- LI, J. et al. Two birds one stone: on both cold-start and long-tail recommendation. In: ACM. *Proceedings of the 25th International Conference on Multimedia*. Mountain View, USA, 2017. p. 898–906.
- LI, L.; TANIAR, D. A taxonomy for distance-based spatial join queries. *International Journal of Data Warehousing and Mining (IJDWM)*, IGI Global, v. 13, n. 3, p. 1–24, 2017.
- LI, M. et al. Efficient processing of location-aware group preference queries. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. Indianapolis, USA: [s.n.], 2016. p. 559–568.
- LIAN, D. et al. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, USA: [s.n.], 2014. p. 831–840.
- LIAO, Z.; TAJIMA, K. Disjunctive sets of phrase queries for diverse query suggestion. In: IEEE. *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. Thessaloniki, Greece, 2019. p. 449–455.
- LIKA, B.; KOLOMVATSOS, K.; HADJIEFTHYMIADES, S. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, Elsevier, v. 41, n. 4, p. 2065–2073, 2014.
- LIU, B. et al. Learning geographical preferences for point-of-interest recommendation. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. Chicago, USA: [s.n.], 2013. p. 1043–1051.
- LIU, J. et al. Clue-based spatio-textual query. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 10, n. 5, p. 529–540, 2017.
- LIU, J. et al. Finding pareto optimal groups: Group-based skyline. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 8, n. 13, p. 2086–2097, 2015.
- LIU, Q. et al. Enhancing collaborative filtering by user interest expansion via personalized ranking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 42, n. 1, p. 218–233, 2011.
- LIU, Y. et al. Probesim: Scalable single-source and top-k simrank computations on dynamic graphs. *Proceedings of the VLDB Endowment*, v. 11, n. 1, 2017.
- LU, H.; YIU, M. L.; XIE, X. Querying spatial data by dominators in neighborhood. *Information Systems*, Elsevier, v. 77, p. 71–85, 2018.
- LUCCHESI, 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*. Pisa, Italy, 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*. Parramatta, Australia, 2015. p. 4.

MAHDAVINEJAD, M. S. et al. Machine learning for internet of things data analysis: A survey. *Digital Communications and Networks*, Elsevier, v. 4, n. 3, p. 161–175, 2018.

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*. Cambridge, United Kingdom: Cambridge university press, 2012.

MANOLOPOULOS, Y.; PAPADOPOULOS, A. N.; VASSILAKOPOULOS, M. G. *Spatial databases: technologies, techniques and trends*. Calgary, Canada: Idea Group Inc (IGI), 2005.

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.

MCCALLUM, A. et al. A comparison of event models for naive bayes text classification. In: CITESEER. *AAAI-98 workshop on learning for text categorization*. Madison, USA, 1998. v. 752, n. 1, p. 41–48.

MCCRAE, J. P. *The Linked Open Data Cloud*. 2019. Available at: <<https://lod-cloud.net/>>. Accessed: 2019-06-13.

MENG, X.; LI, P.; ZHANG, X. A personalized and approximated spatial keyword query approach. *IEEE Access*, IEEE, v. 8, p. 44889–44902, 2020.

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

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

OCHIENG, P. An analysis of the strengths and limitation of qualitative and quantitative research paradigms. *Problems of Education in the 21st Century*, Scientia Socialis Ltd., v. 13, p. 13, 2009.

PAN, J. Z. *Resource description framework*. 2th. ed. Berlin, Germany: Springer, 2009. 71–90 p.

- PAPADIAS, D. et al. Efficient olap operations in spatial data warehouses. In: *International Symposium on Spatial and Temporal Databases (SSTD)*. Redondo Beach, USA: Springer, 2003. p. 443–459.
- PENG, F. et al. Multi-level preference regression for cold-start recommendations. *International Journal of Machine Learning and Cybernetics*, Springer, v. 9, n. 7, p. 1117–1130, 2018.
- PÉREZ, J.; ARENAS, M.; GUTIERREZ, C. Semantics and complexity of sparql. *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*, 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.
- QIAN, C. et al. Geosot-based spatiotemporal index of massive trajectory data. *ISPRS International Journal of Geo-Information*, Multidisciplinary Digital Publishing Institute, v. 8, n. 6, p. 284, 2019.
- QIAN, Z. et al. Semantic-aware top-k spatial keyword queries. *World Wide Web*, Springer, v. 21, n. 3, p. 573–594, 2018.
- QIAO, B. et al. A top-k spatial join querying processing algorithm based on spark. *Information Systems*, Elsevier, v. 87, p. 101419, 2020.
- QIU, F.; CHO, J. Automatic identification of user interest for personalized search. In: *Proceedings of the 15th international conference on World Wide Web*. Edinburgh, Scotland: [s.n.], 2006. p. 727–736.
- QUEIRÓS, A.; FARIA, D.; ALMEIDA, F. Strengths and limitations of qualitative and quantitative research methods. *European Journal of Education Studies*, 2017.
- RATHOD, P.; DESMUKH, S. A personalized mobile search engine based on user preference. In: IEEE. *International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. Chennai, India, 2017. p. 1136–1141.
- RIGAUX, P.; SCHOLL, M.; VOISARD, A. *Spatial databases: with application to GIS*. Burlington, USA: Morgan Kaufmann, 2002.
- ROCHA-JUNIOR, J. B. *Efficient processing of preference queries in distributed and spatial databases*. Thesis (Ph.D.) — Norwegian University of Science and Technology, 2012.
- ROCHA-JUNIOR, J. B. et al. Efficient processing of top-k spatial keyword queries. In: SPRINGER. *International Symposium on Spatial and Temporal Databases*. Minneapolis, USA, 2011. p. 205–222.

- ROCHA-JUNIOR, J. B. et al. Efficient processing of top-k spatial preference queries. *Proceedings of the VLDB Endowment*, VLDB Endowment, v. 4, n. 2, p. 93–104, 2010.
- SALGADO, C.; CHEEMA, M. A.; TANIAR, D. An efficient approximation algorithm for multi-criteria indoor route planning queries. In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Seattle, USA: [s.n.], 2018. p. 448–451.
- 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*. Riobamba City, Ecuador, 2018. p. 283–300.
- SARKER, M. K. et al. Explaining trained neural networks with semantic web technologies: First steps. *Proceedings of the Twelfth International Workshop on Neural-Symbolic Learning and Reasoning*, 2017.
- SARWAT, M. et al. Lars*: An efficient and scalable location-aware recommender system. *Transactions on Knowledge & Data Engineering*, IEEE, n. 6, p. 1–1, 2014.
- 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*. Houston, USA, 2018. p. 1557–1570.
- SINOARA, R. A. et al. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, Elsevier, v. 163, p. 955–971, 2019.
- SKORKOVSKÁ, L. Relevant documents selection for blind relevance feedback in speech information retrieval. In: SPRINGER. *International Conference on Text, Speech, and Dialogue*. Brno, Czech Republic, 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.
- SONG, J. J. et al. An effective recall-oriented information retrieval system evaluation. In: SPRINGER. *International Conference on Big Data Applications and Services*. Tashkent, Uzbekistan, 2017. p. 43–49.
- 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.

TSATSANIFOS, G.; VLACHOU, A. On processing top-k spatio-textual preference queries. In: *18th International Conference on Extending Database Technology (EDBT)*. Brussels, Belgium: [s.n.], 2015. p. 433–444.

VANBELLE, S. A new interpretation of the weighted kappa coefficients. *Psychometrika*, Springer, v. 81, n. 2, p. 399–410, 2016.

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

XIA, Y.; GONG, J.; ZHU, X. Personalized retrieval of spatial information combining user profile with query request. In: IEEE. *19th International Conference on Geoinformatics*. Shanghai, China, 2011. p. 1–4.

XIE, M. et al. Learning graph-based poi embedding for location-based recommendation. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. Indianapolis, USA: [s.n.], 2016. p. 15–24.

XU, J.; CROFT, W. B. Query expansion using local and global document analysis. In: ACM. *SIGIR Forum*. New York, USA, 2017. v. 51, n. 2, p. 168–175.

XU, S. Bayesian naïve bayes classifiers to text classification. *Journal of Information Science*, SAGE Publications Sage UK: London, England, v. 44, n. 1, p. 48–59, 2018.

XU, S. et al. Exploring folksonomy for personalized search. In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. Singapore: [s.n.], 2008. p. 155–162.

YANG, C. et al. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Halifax, Canada: [s.n.], 2017. p. 1245–1254.

YE, M. et al. Exploiting geographical influence for collaborative point-of-interest recommendation. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. Beijing, China: [s.n.], 2011. p. 325–334.

- YIU, M. L. et al. Top-k spatial preference queries. In: IEEE. *Proceedings of the 23rd International Conference on Data Engineering Workshop (ICDE)*. Washington, USA, 2007. p. 1076–1085.
- YIU, M. L. et al. Ranking spatial data by quality preferences. *Transactions on Knowledge and Data Engineering*, IEEE, v. 23, n. 3, p. 433–446, 2010.
- YIU, M. L. et al. Ranking spatial data by quality preferences. *Transactions on Knowledge and Data Engineering*, IEEE, v. 23, n. 3, 2011.
- ZACHARATOU, E. T. et al. Efficient bundled spatial range queries. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Chicago, USA: [s.n.], 2019. p. 139–148.
- ZARRINKALAM, F.; KAHANI, M. A multi-criteria hybrid citation recommendation system based on linked data. In: IEEE. *2nd International eConference on Computer and Knowledge Engineering (ICCKE)*. Mashhad, Iran, 2012. p. 283–288.
- ZEMEDE, B. A.; GAO, B. J. Personalized search with editable profiles. In: IEEE. *International Conference on Big Data*. Boston, USA, 2017. p. 4872–4874.
- ZENG, W. et al. Tup-rs: Temporal user profile based recommender system. In: SPRINGER. *International Conference on Artificial Intelligence and Soft Computing*. Zakopane, Poland, 2018. p. 463–474.
- ZHANG, C. et al. Gmove: Group-level mobility modeling using geo-tagged social media. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, USA: [s.n.], 2016. p. 1305–1314.
- ZHANG, S. et al. A caching and spatial k-anonymity driven privacy enhancement scheme in continuous location-based services. *Future Generation Computer Systems*, Elsevier, v. 94, p. 40–50, 2019.
- ZHANG, S. et al. Exploiting ranking consistency principle in representation learning for location promotion. In: SPRINGER. *International Conference on Database Systems for Advanced Applications*. Gold Coast, Australia, 2018. p. 457–473.
- ZHANG, X. et al. Dualds: A dual discriminative rating elicitation framework for cold start recommendation. *Knowledge-Based Systems*, Elsevier, v. 73, p. 161–172, 2015.
- ZHIMING, C.; AREFIN, M. S.; MORIMOTO, Y. Skyline queries for spatial objects: a method for selecting spatial objects based on surrounding environments. In: IEEE. *Third International Conference on Networking and Computing*. Okinawa, Japan, 2012. p. 215–220.
- ZHOU, Y. et al. Hybrid index structures for location-based web search. In: ACM. *Proceedings of the 14th International Conference on Information and Knowledge Management*. Bremen, Germany, 2005. p. 155–162.

ZHU, G. et al. Hymj: A hybrid structure-aware approach to distributed multi-way join query. In: IEEE. *35th International Conference on Data Engineering (ICDE)*. Macau, China, 2019. p. 1726–1729.

ZHU, H.; LIU, F.; LI, H. Efficient and privacy-preserving polygons spatial query framework for location-based services. *Internet of Things Journal*, IEEE, v. 4, n. 2, p. 536–545, 2017.

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

ZHU, W.-Y. et al. Modeling user mobility for location promotion in location-based social networks. In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. Sydney, Australia: [s.n.], 2015. p. 1573–1582.

ZOBEL, J.; MOFFAT, A. Inverted files for text search engines. *Computing Surveys*, ACM, New York, USA, v. 38, n. 2, p. 6, 2006.

Appendix

A

USER PROFILE BASED ON ROOM ASPECT VALUE

This appendix presents the user profile based on the *room* aspect rating from the Opin-Rank dataset. It follows the .arff file structure to be read by the Weka framework. The first line describes the name of the data. Then, the reserved word “class” defines the review labels (one and five) and the data type of the instances (string). After the tag data each label is followed by the user review describing a preference for hotels with room quality. The profile contains ten reviews describing hotels with good room aspect rating and ten reviews describing the worst hotels for this particular user. This profile data and the other profiles employed in this thesis are available online at <<http://tiny.cc/i3babz>>.

@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 facilities 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 laptop 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 executivite/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 sumptuously 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 & 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 occasions, and both times have been completely 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 enough 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 & 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

& Spa has it all, 400%! We travelled with my parents and had an interconnecting Prestige Room & 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: sumptuous!!! 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 don't 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 1st 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 Apts, which weren't much good either but a slight improvement. This is a very mediocre 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 wasn't 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 unboxed, unwrapped bunch of facial tissue. Imagine the joy! Apparently, housekeeping was gone for the night and no one 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 they're 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.... you'll 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 didn't seem sanitary, so we passed on the breakfast offerings during our stay. The Ramee Group is a pretty large, 'middle market' chain, and I've stayed at an Ramee in Bahrain (minus the negative experience),

but clearly, the quality 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 separate 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 didn't 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 didn't even provide extra linen for the children. (I'm 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 normally 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 transferred 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 Sheikh 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 so 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 16,000 AED per month) and don't 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 can't 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 to 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.'

RATINGS AND QUERIES RELATED TO THE ASPECT RATING ROOM

This appendix is reserved to describe the ratings and queries related to the aspect rating *room* in the Opinrank dataset. The first line of the file defines the aspect rating to which the file is related. Then, six queries related to the aspect rating are described. These ratings are based on real-user queries when searching for hotels with high room quality. Ganesan and Zhai (2011) describe the methodology to define these queries. After the tag `#judge`, each line contains a hotel name and its respect average rating value. This dataset is also available online at <https://bit.ly/3bLRKjm>.

#cat=room,

#query=nice room;9658

#query=great room;9659

#query=cozy rooms;9660

#query=spacious room;9661

#query=comfy room;9662

#query=comfortable room;9663

#judge

are_dubai_residence_and_spa_at_one_and_only_royal_mirage;4.95

are_dubai_burj_al_arab;4.898936170212766

are_dubai_dar_al_masyaf_at_madinat_jumeirah;4.890243902439025

are_dubai_oasis_beach_tower_apartments;4.861538461538461

are_dubai_al_maha_desert_resort;4.857142857142857

are_dubai_villa_rotana_dubai;4.833333333333333

are_dubai_arjaan_dubai_media_city;4.833333333333333

are_dubai_mina_a_salam_at_madinat_jumeirah;4.826530612244898

are_dubai_the_palace_the_old_town;4.815789473684211

are_dubai_burjuman_arjaan_dubai;4.814814814814815

are_dubai_raffles_dubai;4.8108108108108105

are_dubai_grosvenor_house_west_marina_beach_by_le_meridien_dubai;4.810344827586207

are_dubai_bonnington_jumeirah_lakes_towers;4.8076923076923075

are_dubai_al_qasr_at_madinat_jumeirah;4.806451612903226

are_dubai_crowne_plaza_dubai_festival_city;4.784810126582278

are_dubai_grand_hyatt_dubai;4.780701754385965

are_dubai_desert_palm_resort_spa;4.777777777777778

are_dubai_intercontinental_dubai_festival_city;4.776119402985074

are_dubai_westin_dubai_mina_seyahi_beach_resort_marina;4.763313609467455

are_dubai_xclusive_hotel_apartments;4.75

are_dubai_one_only_royal_mirage;4.742424242424242

are_dubai_jumeirah_beach_hotel;4.73421926910299

are_dubai_le_royal_meridien_beach_resort_spa;4.714285714285714
are_dubai_symphony_hotel_apartments;4.666666666666667
are_dubai_four_points_by_sheraton_downtown_dubai;4.650485436893204
are_dubai_al_manzil_hotel;4.64957264957265
are_dubai_grand_hyatt_hotel_dubai;4.636363636363637
are_dubai_arabian_court_at_one_and_only_royal_mirage;4.625
are_dubai_jebel_ali_palm_tree_court_spa;4.622950819672131
are_dubai_jumeira_rotana;4.621621621621622
are_dubai_kempinski_hotel_mall_of_the_emirates;4.592592592592593
are_dubai_qamardeen_hotel;4.551020408163265
are_dubai_park_hyatt_dubai;4.542857142857143
are_dubai_the_palace_at_one_only_royal_mirage;4.538461538461538
are_dubai_shangri_la_hotel;4.538461538461538
are_dubai_hilton_dubai_creek;4.514925373134329
are_dubai_taj_palace_hotel;4.509803921568627
are_dubai_four_points_by_sheraton_sheikh_zayed_road_dubai;4.509615384615385
are_dubai_hatta_fort_hotel;4.5
are_dubai_the_ritz_carlton_dubai;4.48
are_dubai_the_address_downtown_burj_dubai;4.46875
are_dubai_the_monarch_dubai;4.466666666666667
are_dubai_traders_hotel_dubai;4.462962962962963
are_dubai_jumeirah_emirates_towers_hotel;4.46
are_dubai_copthorne_hotel_dubai;4.454545454545454
are_dubai_le_meridien_al_sondos_suites;4.434782608695652
are_dubai_star_boutique_hotel_apartment;4.428571428571429
are_dubai_rimal_rotana_dubai;4.418181818181818
are_dubai_hilton_dubai_jumeirah;4.412639405204461
are_dubai_the_fairmont_dubai;4.395833333333333
are_dubai_jumeirah_bab_al_shams_desert_resort_spa;4.318840579710145
are_dubai_le_meridien_mina_seyahi_beach_resort_and_marina;4.316129032258065
are_dubai_chelsea_tower_hotel_apartments;4.3125

are_dubai_hyatt_regency_dubai;4.3076923076923075
are_dubai_arabian_dreams_hotel_apartments;4.3076923076923075
are_dubai_holiday_inn_dubai_al_barsha;4.3055555555555555
are_dubai_dusit_thani_dubai;4.290322580645161
are_dubai_courtyard_by_marriott_green_community_dubai;4.285714285714286
are_dubai_coral_deira_dubai;4.2727272727272725
are_dubai_sheraton_dubai_creek_hotel_and_towers;4.260869565217392
are_dubai_rihab_rotana_dubai;4.25
are_dubai_zagy_arabian_suites;4.25
are_dubai_renaissance_dubai_hotel;4.2153846153846155
are_dubai_atlantis_the_palm;4.212624584717608
are_dubai_habtoor_grand_resort_spa;4.205128205128205
are_dubai_al_bustan_rotana_dubai;4.2
are_dubai_le_meridien_fairway;4.2
are_dubai_grand_millennium_dubai;4.2
are_dubai_arabian_courtyard_hotel_spa;4.186770428015564
are_dubai_jw_marriott_hotel_dubai;4.176470588235294
are_dubai_tamani_hotel_marina;4.176470588235294
are_dubai_dubai_international_hotel;4.166666666666667
are_dubai_al_murooj_rotana_dubai;4.159090909090909
are_dubai_towers_rotana_dubai;4.148936170212766
are_dubai_rolla_residence;4.142857142857143
are_dubai_le_meridien_dubai;4.137931034482759
are_dubai_holiday_inn_express_dubai_safa_park;4.133333333333334
are_dubai_novotel_deira_city_centre;4.130434782608695
are_dubai_radisson_blu_hotel_dubai_deira_creek;4.086956521739131
are_dubai_crowne_plaza_hotel_dubai;4.085106382978723
are_dubai_angsana_dubai;4.083333333333333
are_dubai_majestic_hotel;4.081632653061225
are_dubai_jebel_ali_golf_resort_spa;4.08
are_dubai_moevenpick_hotel_bur_dubai;4.076923076923077

are_dubai_royal_ascot_hotel;4.074074074074074
are_dubai_flora_grand_hotel;4.0
are_dubai_ramada_dubai;4.0
are_dubai_metropolitan_palace_hotel;4.0
are_dubai_radisson_blu_hotel_dubai_media_city;4.0
are_dubai_golden_sands_hotel_apartments;3.951388888888889
are_dubai_sheraton_jumeirah_beach_resort_towers;3.916666666666665
are_dubai_le_meridien_residence_deira;3.909090909090909
are_dubai_holiday_inn_express_dubai_jumeirah;3.8947368421052633
are_dubai_dhow_palace_hotel;3.8947368421052633
are_dubai_grandeur_hotel;3.888888888888889
are_dubai_rydges_plaza_dubai;3.878787878787879
are_dubai_novotel_world_trade_centre_dubai;3.847222222222223
are_dubai_arabian_park_hotel;3.8035714285714284
are_dubai_riviera_hotel;3.782608695652174
are_dubai_hawthorn_hotel_deira;3.769230769230769
are_dubai_holiday_inn_express_dubai_internet_city;3.722222222222223
are_dubai_ibis_deira_city_centre;3.6315789473684212
are_dubai_four_points_by_sheraton_bur_dubai;3.622222222222222
are_dubai_metropolitan_hotel_deira;3.5789473684210527
are_dubai_ibis_world_trade_centre_dubai;3.5434782608695654
are_dubai_dream_palace_hotel;3.526315789473684
are_dubai_regent_beach_resort;3.5
are_dubai_hotel_eureka;3.5
are_dubai_highland_hotel;3.48
are_dubai_grand_midwest_hotel_apartments;3.473684210526316
are_dubai_dubai_marine_beach_resort_and_spa;3.4285714285714284
are_dubai_avari_dubai_hotel;3.416666666666665
are_dubai_chelsea_hotel;3.416666666666665
are_dubai_marco_polo_hotel;3.416666666666665
are_dubai_st_george_hotel_dubai;3.4

are_dubai_sheraton_deira_hotel;3.3684210526315788
are_dubai_dubai_nova_hotel;3.357142857142857
are_dubai_versailles_hotel;3.3333333333333335
are_dubai_coral_oriental_dubai;3.272727272727273
are_dubai_grand_moov_hotel;3.2244897959183674
are_dubai_ascot_hotel;3.2
are_dubai_metropolitan_hotel_dubai;3.19672131147541
are_dubai_jormand_hotel_apartments;3.1818181818181817
are_dubai_lotus_boutique_hotel;3.1818181818181817
are_dubai_landmark_plaza_hotel;3.1538461538461537
are_dubai_millennium_airport_hotel_dubai;3.142857142857143
are_dubai_le_meridien_dar_al_sondos_hotel_apartments;3.125
are_dubai_ramee_apartment_hotel;3.1052631578947367
are_dubai_seaview_hotel;3.0952380952380953
are_dubai_regal_plaza_dubai;3.0625
are_dubai_landmark_hotel;3.0
are_dubai_san_marco_hotel;2.9523809523809526
are_dubai_k_porte_inn_hotel;2.9375
are_dubai_the_carlton_tower_hotel;2.909090909090909
are_dubai_fortune_hotel;2.9
are_dubai_city_centre_hotel_residence;2.857142857142857
are_dubai_panorama_hotel_bur_dubai;2.8181818181818183
are_dubai_pearl_residence;2.8
are_dubai_york_international_hotel;2.7857142857142856
are_dubai_holiday_inn_downtown_dubai;2.761904761904762
are_dubai_admiral_plaza_hotel;2.7586206896551726
are_dubai_nihal_hotel;2.7142857142857144
are_dubai_regent_palace_hotel;2.619047619047619
are_dubai_lotus_hotel_dubai;2.5454545454545454
are_dubai_queens_hotel;2.4545454545454546
are_dubai_palm_beach_rotana_inn;2.4166666666666665

are_dubai_orchid_hotel;2.357142857142857

are_dubai_panorama_deira;2.1

Appendix

C

SURVEY SUBMITTED TO COLLECT OPINIONS FROM SPECIALISTS ABOUT THE PROTOTYPE

This appendix contains a copy of the online survey used to obtain feedback from specialists about the use case COVID-19 Geo-monitor, earlier addressed in Chapter 6.

Geo-Monitor para COVID-19

A pandemia de COVID-19 impõe a necessidade de isolar determinadas áreas urbanas para conter a propagação do vírus. Neste protótipo, apresentamos uma ferramenta capaz de exibir no mapa os locais com um alto número de infectados, descrevendo em detalhes os pontos vizinhos a esses pacientes. Além disto, é possível informar ao paciente qual a Unidade Básica de Saúde (UBS) mais próxima de sua localização.

Este formulário tem como objetivo coletar informações sobre o protótipo desenvolvido. As informações coletadas são anônimas e serão utilizadas na produção de artigos ou na tese final de doutorado.

Nós agradecemos bastante a sua contribuição!

* Required

1. Email address *

2. Antes de começar, precisamos que você descreva a sua função na sua organização. Por exemplo, "eu desenvolvo sistema que faz x", ou "eu conduzo pesquisa em y". *

Coleta de Opinião

A seguir, coletaremos a sua opinião sobre o nosso protótipo. Indique na escala de 1 a 5 o quão útil você considera as funcionalidades apresentadas. Se possível, descreva como podemos melhorar essas funcionalidades.

3. Você acredita que a funcionalidade apresentada para associar um paciente à UBS mais próxima de sua residência é útil e pode resolver problemas reais? *

Mark only one oval.

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Útil

4. Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

5. Você acredita que a funcionalidade apresentada para ranquear locais considerando casos positivos de COVID-19 na vizinhança é útil e pode contribuir com as autoridades para o controle da pandemia? *

Mark only one oval.

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Útil

6. Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

7. Uma das contribuições de nossa pesquisa neste protótipo é melhorar a descrição textual dos locais no mapa utilizando bases de dados externas. Assim, é possível considerar na análise locais que não seriam analisados devido a falta de descrição. Você concorda que essa melhoria possibilita identificar locais vulneráveis (e.g. escolas, abrigos para idosos) que possuem muitos casos positivos ao seu redor? *

Mark only one oval.

	1	2	3	4	5	
Discordo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo totalmente

8. Se você discorda, poderia justificar o motivo?

9. Quais das seguintes palavras melhor descreve a sua sensação ao protótipo apresentado? *

Check all that apply.

- Excitado (a)
 Satisfeito (a)
 Neutro (a)
 Confuso (a)
 Insatisfeito (a)
 Irritado (a)

Other: _____

Appendix

D

ANONYMIZED RESPONSES FROM SPECIALISTS ABOUT THE PROTOTYPE

This appendix is a copy of all responses provided by the specialists in their raw form. All responses are anonymous.

Você acredita que a funcionalidade apresentada para associar um paciente à UBS mais próxima de sua residência é útil e pode resolver problemas reais? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

Seria interessante implementar uma função para buscar a localização atual do usuário e preencher no box de busca para funcionalidade de encontrar USB mais próxima.

Você acredita que a funcionalidade apresentada para ranquear locais considerando casos positivos de COVID-19 na vizinhança é útil e pode contribuir com as autoridades para o controle da pandemia? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

Uma das contribuições de nossa pesquisa neste protótipo é melhorar a descrição textual dos locais no mapa utilizando bases de dados externas. Assim, é possível considerar na análise locais que não seriam analisados devido a falta de descrição. Você concorda que essa melhoria possibilita identificar locais vulneráveis (e.g. escolas, abrigos para idosos) que possuem muitos casos positivos ao seu redor? *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Se você discorda, poderia justificar o motivo?

.....

Quais das seguintes palavras melhor descreve a sua sensação ao protótipo apresentado? *

Excitado (a)

Satisfeito (a)

Neutro (a)

Confuso (a)

Insatisfeito (a)

Irritado (a)

Other:

This content is neither created nor endorsed by Google.

Google Forms

Você acredita que a funcionalidade apresentada para associar um paciente à UBS mais próxima de sua residência é útil e pode resolver problemas reais? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

Você acredita que a funcionalidade apresentada para ranquear locais considerando casos positivos de COVID-19 na vizinhança é útil e pode contribuir com as autoridades para o controle da pandemia? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

Acho que os lugares também poderiam ser ranqueados pela taxa de incidência, além do número de casos

Uma das contribuições de nossa pesquisa neste protótipo é melhorar a descrição textual dos locais no mapa utilizando bases de dados externas. Assim, é possível considerar na análise locais que não seriam analisados devido a falta de descrição. Você concorda que essa melhoria possibilita identificar locais vulneráveis (e.g. escolas, abrigos para idosos) que possuem muitos casos positivos ao seu redor? *

Discordo totalmente 1 2 3 4 5 Concordo totalmente

Se você discorda, poderia justificar o motivo?

.....

Quais das seguintes palavras melhor descreve a sua sensação ao protótipo apresentado? *

- Excitado (a)
- Satisfeito (a)
- Neutro (a)
- Confuso (a)
- Insatisfeito (a)
- Irritado (a)
- Other:

This content is neither created nor endorsed by Google.

Google Forms

Você acredita que a funcionalidade apresentada para associar um paciente à UBS mais próxima de sua residência é útil e pode resolver problemas reais? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

Não identifiquei limitação na funcionalidade.

Você acredita que a funcionalidade apresentada para ranquear locais considerando casos positivos de COVID-19 na vizinhança é útil e pode contribuir com as autoridades para o controle da pandemia? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

Uma sugestão é permitir fazer um ranqueamento sem a necessidade de definir o tipo de lugar permitindo a análise de múltiplos estabelecimentos ao mesmo tempo (bares, hospitais, escolas, academias, unidades básicas de saúde).

Uma das contribuições de nossa pesquisa neste protótipo é melhorar a descrição textual dos locais no mapa utilizando bases de dados externas. Assim, é possível considerar na análise locais que não seriam analisados devido a falta de descrição. Você concorda que essa melhoria possibilita identificar locais vulneráveis (e.g. escolas, abrigos para idosos) que possuem muitos casos positivos ao seu redor? *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Se você discorda, poderia justificar o motivo?

.....

Quais das seguintes palavras melhor descreve a sua sensação ao protótipo apresentado? *

Excitado (a)

Satisfeito (a)

Neutro (a)

Confuso (a)

Insatisfeito (a)

Irritado (a)

Other:

This content is neither created nor endorsed by Google.

Google Forms

Você acredita que a funcionalidade apresentada para associar um paciente à UBS mais próxima de sua residência é útil e pode resolver problemas reais? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

A limitação existente precede a funcionalidade e reside na disponibilidade de dados geocodificados.

Você acredita que a funcionalidade apresentada para ranquear locais considerando casos positivos de COVID-19 na vizinhança é útil e pode contribuir com as autoridades para o controle da pandemia? *

	1	2	3	4	5	
Inútil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	Útil

Você identificou alguma limitação na funcionalidade anterior? Como seria possível melhorar esta funcionalidade?

A limitação existente precede a funcionalidade e reside na disponibilidade de dados geocodificados.

Uma das contribuições de nossa pesquisa neste protótipo é melhorar a descrição textual dos locais no mapa utilizando bases de dados externas. Assim, é possível considerar na análise locais que não seriam analisados devido a falta de descrição. Você concorda que essa melhoria possibilita identificar locais vulneráveis (e.g. escolas, abrigos para idosos) que possuem muitos casos positivos ao seu redor? *

1 2 3 4 5

Discordo totalmente Concordo totalmente

Se você discorda, poderia justificar o motivo?

.....

Quais das seguintes palavras melhor descreve a sua sensação ao protótipo apresentado? *

Excitado (a)

Satisfeito (a)

Neutro (a)

Confuso (a)

Insatisfeito (a)

Irritado (a)

Other:

This content is neither created nor endorsed by Google.

Google Forms