Journal of Universal Computer Science, vol. 24, no. 11 (2018), 1561-1581
submitted: 28/2/18, accepted: 7/9/18, appeared: 28/11/18 © J.UCS

Enhancing Spatial Keyword Preference Query with Linked Open Data

João Paulo Dias de Almeida
(Federal University of Bahia, Brazil
joao.dias@ufba.br)

Frederico Araújo Durão
(Federal University of Bahia, Brazil
fdurao@ufba.br)

Arthur Fortes da Costa
(University of São Paulo, Brazil
fortes@icmc.usp.br)

Abstract: This paper presents a Spatial Keyword Preference Query (SKPQ) enhanced by Linked Open Data. This query selects objects based on the textual description of features in their neighborhood. The spatial relationship between objects and features is explored by the SKPQ using a Spatial Inverted Index. In our approach, the spatial relationship is explored using SPARQL. However, the main benefit of using SPARQL is obtained by measuring the textual relevance between features' description and user's keywords. The object description in Linked Open Data is much richer than traditional spatial databases, which leads to a more precise similarity measure than the one employed in the traditional SKPQ. We present an enhanced SKPQ, an algorithm to process this enhanced query, and two experimental evaluations of the proposed algorithm, comparing it with the traditional SKPQ. The first conducted experiment indicate a relative NDCG improvement of the proposed approach over the traditional SKPQ of 20% when using random query keywords. The second experiment shows that using real query keywords, our approach obtained a significant increase in the MAP score.

Key Words: Spatial data, Query evaluation, Query processing, Linked Open Data
Category: H.3, H.3.3, H.3.5

1 Introduction

The advances in the location-aware hardware and software technologies stimulate the development of location-based services (i.e. Foursquare services). Location-based services enable their users to describe, rate and interact with urban spaces. In this context, any data associated with spatial coordinates are named spatial data or spatial object [Cao et al. 2012]. Location-based services can access a spatial database to select objects that satisfy the user's preference. In order to access the spatial object, the Location-based service can employ spatial preference queries. Instead of receiving a small or possibly huge and unordered result set, preference queries offer a manageable set of "best" answers, which satisfy the query best. Many preference queries specify the user preference using query keywords. For instance, a user looking for a Japanese restaurant can specify his preference with the query keywords "japanese restaurant".

Preference queries which use keywords evaluate the object whose textual description shares words in common with the query keywords as relevant for the user [Cao et al. 2012, Cong et al. 2009]. In this way, the more words in common, the better the textual relevance between an object and the query keywords. However, this evaluation method has limitations, especially to objects with short textual descriptions. For example, suppose a spatial area (e.g. a city) where two spatial objects are located. The query keywords are "japanese restaurant" and each object has one textual description represented by the following strings: "oriental food" and "cinema". This query is not able to return any object because neither the word "japanese" nor "restaurant" are present in any textual description. One possible solution for this problem is offering a wider textual description for the spatial objects. The object described as "oriental food" could be a Japanese restaurant but we can not be sure because of its poor textual description.

Motivated by this problem, we use the data available at Linked Open Data (LOD) cloud to enrich the textual description of objects. A large number of researches have recently studied how to improve the object's textual description using the LOD cloud. This improvement is applied in several areas of research, such as Recommender Systems [Hegde et al. 2011, Fernández-Tobías et al. 2011] and Information Retrieval [Karam and Melchiori 2013, Becker and Bizer 2009]. However, to the best of our knowledge, we are the first to apply a similar improvement in a Spatial Keyword Preference query.

This paper proposes a location-based solution that exploits the benefits of a LOD dataset for enriching the object textual description. We employ our solution at Top-k Spatial Keyword Preference Query (SKPQ) [de Almeida and Rocha-Junior 2016]. This query accesses objects from a traditional database like OpenStreetMap. However, a LOD database like DBpedia contains objects' descriptions wider than the ones available at OpenStreetMap. The contributions of this work are i) a novel semantic model for enhancing the SKPQ, ii) an algorithm to process the SKPQ with a LOD dataset, iii) an analysis on how the wider textual description influences the query results.

The remainder of this paper is structured as follows: Section 2 introduces a motivating scenario; Section 3 and 4 presents the related work and background; then Section 5 describes the algorithm used to process SKPQ using a LOD dataset; Section 6 and 7 contain the experimental evaluations and the discussion about these evaluations, finally, Section 8 presents the conclusions and future work.

2 Motivating Scenario

Several queries are processed using the Vector Space Model (VSM) to evaluate the textual relevance between query keywords and object's textual description [de Almeida and Rocha-Junior 2016], [Cao et al. 2012], [Cong et al. 2009]. The VSM indicates that two strings are textual relevant when they share words. The Top-k Spatial Keyword Preference Query (SKPQ) is a preference query that uses query keywords to describe the user preference and is processed using VSM. The SKPQ searches for spatial objects of user's interest based on spatio-textual objects[1] of reference (features) in their spatial neighborhood. For example, Figure 1 describes a spatial area with spatial objects p (e.g. hotels) and features f (e.g. any establishment). Consider a user interested in book a hotel close to a Japanese restaurant. The user specifies the query keywords "japanese restaurant" and the spatial selection criteria (represented by the circle around the objects p). An evaluation method defines that the textual description of the object f_1 "restaurant" has textual relevance to query keywords. However, the textual description of object f_4 "japanese restaurant" is more textual relevant because it has the same words as the query keywords. Objects f_2, f_3, f_5, f_6, f_7 have no textual relevance to the query keyword, while f_5 does not satisfy the spatial selection criteria too. The SKPQ returns the object p_3 as the best hotel for the user's need, since f_4 has the greatest textual relevance among all features and satisfies the spatial selection criteria.

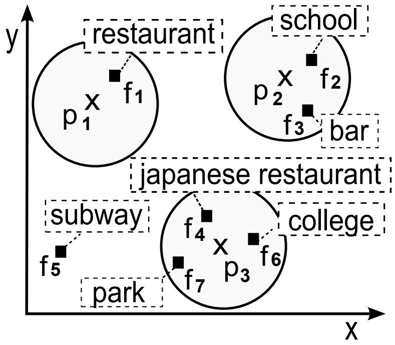


Figure 1: Spatial objects of interest (p) and features (f) associated with their textual descriptions

Suppose a SKPQ with query keywords "oriental food". Considering Figure

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

1, this query does not return any objects. Neither the word “oriental” or “food” are present in any textual description. Note that “oriental food” has semantic relevance to “japanese restaurant”, but the evaluation method is not able to identify this relationship. In this example, the query fails to retrieve relevant objects when query keywords are“oriental food”. So, we propose a solution using a LOD dataset to enhance the object textual description, in order to achieve better object evaluation. A wider textual description for objects f can improve the object evaluation. If object f_4 had a better textual description, the word “food” or “oriental” might appear in the textual description. In this scenario, the semantic relationship offered by the LOD dataset can be very helpful too.

3 Related Work

Several studies suggest that LOD datasets can be used to improve textual descriptions of objects of user’s interest. Hegde et al. [Hegde et al. 2011] described an augmented reality browser that uses LOD to enhance the description of objects, offering a better recommendation. The objects were represented by a semantic relationship between them and several spatial data repositories such as Wikipedia[2] and YouTube. Using Natural Language Processing techniques, the user’s profile is semantically related to a point of interest (POI). Then the personalized set of POIs is delivered to the user. Similarly, Karam and Melchiori [Karam and Melchiori 2013] presented a way to improve POIs description quality using LOD. They developed the M-PREGeD, a conceptual framework aiming to improve the accuracy of spatial data from different LOD sources. In M-PREGeD, voluntary users can generate or update POIs descriptions in order to enhance it. Aiming the same goal, we use DBpedia to enhance the textual description of features. However, we do not make use of voluntary users to help the process because we aim for an automatic enhancement approach.

The popularization of GPS (Global Positioning System) enabled devices increases significantly the volume of spatial data produced in the last years. This phenomenon stimulates new systems making use of spatial data associated with LOD. Fernández-Tobías et al. [Fernández-Tobías et al. 2011] made use of LOD and spatial data to recommend musicians related to the architecture around the user’s spatial position. Likewise our approach, they used LOD to obtain data about a spatial area (ex: architecture in Rome) but they did not make use of any spatial information (ex: latitude or longitude) in their recommendation. We use the spatial information to select objects that satisfy the user’s information need. Equally important, Becker and Bizer [Becker and Bizer 2009] presented 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

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

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

Accordingly Braun et al. [Braun et al. 2010], simple text description hinders the extraction of relations between objects. In order to mitigate this problem, Braun et al. propose a semantic representation of objects using LOD. They created a collaborative spatial database compounded by POIs. In this database, users can define the ontology category of each POI. To improve the POI quality, a revision engine based on data mining techniques is provided.

Meta-Knowledge is another approach employed to enrich textual description. Meta-Knowledge refers to include metadata at textual corpus using an annotation scheme. For example, a news text about an event can include metadata like the modality, subjectivity, source, polarity and specificity of the event [Thompson et al. 2017]. This approach enriches the metadata instead of the data describing the object. In this work, we aim to enrich the data describing a spatial object.

4 Background

4.1 Linked Open Data

Accordingly to [Becker and Bizer 2009], the Web has evolved into a space where both documents and data are linked. In order to support this new Web, a set of practices for publishing and connect structured data has been proposed by Berners-Lee [Berners-Lee 2011]. This set of practices is known as Linked Data because it enables a user to start browsing in one data source and then navigate along links into related data sources.

In a nutshell, Linked Data relies on these three technologies: Uniform Resource Identifiers (URIs) [Berners-Lee et al. 2005], the HyperText Transfer Protocol (HTTP) [Fielding et al. 1999], and the Resource Description Framework (RDF) model. A simple way to create linked data is using one RDF file with a URI which points into another file. Suppose an RDF file, named `http://example.org/Hotels`, where hotels around the world are described. Local identifiers (#Venice, #Italy and #Hotel_Danieli) are used to describe one hotel (resource). In Listing 1, hotel Danieli is described with RDF. An HTTP URI `http://example.org/Hotels/#Hotel_Danieli` can be assigned, enabling anyone on the Web to access the hotel's description. When this data is released under an open license it is called Linked Open Data (LOD). In this work, we use two LOD sources: DBpedia and LinkedGeoData.

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

Listing 1: Description of hotel Danieli in an RDF file

4.2 SPARQL

SPARQL is a query language that can be used to express queries across diverse data sources. The data queried using SPARQL might be stored natively as RDF or viewed as RDF via middleware. A SPARQL endpoint is used to enable users to query a knowledge base via the SPARQL query language. DBpedia and LinkedGeoData endpoints can be accessed at http://dbpedia.org/snorql/ and http://linkedgeodata.org/sparql. Listing 2 introduces a SPARQL query to obtain features within 200 m from an object of interest. In Listing 2, *objectURI* is a URI to an object of interest.

The predicate *geo:geometry* is defined at Geo-SPARQL [Perry and Herring 2012], an ontology that represents features and geometries. In Listing 2, the variable *location* matches with the spatial coordinates of objects around an object of interest. The function *bif:st_intersects()* returns true if there is at least one point in common between the spatial coordinates *location* and *sourcegeo*. The tolerance for the matching in units of linear distance is supplied at the third parameter of *bif:st_intersects()*. The tolerance is 200 m as illustrated at Listing 2.

5 Proposed Algorithm

In this section, we present the proposed algorithm to process the SKPQ-LD. This algorithm employs SPARQL to obtain the textual description for features. The traditional SKPQ uses a Spatial Inverted Index (S2I) to index a text file with all textual descriptions needed. Before presenting the algorithm, we describe the S2I structure and how SPARQL was used to obtain the textual description.

S2I is a hybrid index structure able to search spatio-textual data in an optimized way [Rocha-Junior et al. 2011]. Similarly to an Inverted File [Zobel and Moffat 2006], the S2I stores for each term of a vocabulary, one set of objects that contains the term. However, in a different fashion from an Inverted File, the S2I stores the most frequent terms in an aR-tree [Papadias et al. 2001], while the less frequent terms are stored in an Inverted List. Each object has one entry in S2I with object identification, object spatial location, and the term impact.

Term impact is the textual relevance of a term in a document, ignoring other documents of the collection [Salton and Buckley 1988].

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

Listing 2: 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 3: SPARQL query to obtain textual description for one feature

In traditional SKPQ, the textual description of a feature is obtained from S2I. Given a spatial location and one term (keyword), the S2I returns one list with all features that satisfy both the textual relevance and spatial selection criteria. In this work, we query data from the LOD cloud with two objectives: 1) to find features that satisfy the spatial selection criteria, and 2) to obtain their textual description. Listing 2 describes the SPARQL query used to achieve the first objective. While the SPARQL query used to accomplish the second objective is described in Listing 3. More specifically, we access the DBpedia endpoint to read the abstract and comment properties values of each feature obtained. *referenceObjectURI* in Listing 3 is the URI to a feature.

5.1 The Algorithm

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

Algorithm 1 presents the algorithm to process the SKPQ-LD. It receives as input the SKPQ $Q = \{Q.D, Q.r, Q.k\}$, where $Q.D$ is the query keywords, $Q.r$

Algorithm 1: Processing SKPQ-LD

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

is the radius that defines the spatial selection criteria, and $Q.k$ is the number of expected results. The algorithm computes the score of each object $p \in P$ (lines 2-17). Initially, the score of p is zero (line 3). Then, an iterator (line 4) is employed to access all objects f in the spatial vicinity of p. The textual description of each object f is accessed (line 6), and the textual relevance between this description and the query keywords is computed (line 7) using cosine similarity. In this article, we use cosine similarity because we want the term frequency to be determinant over the document length [Zobel and Moffat 2006]. The $getAbstract(iterator.next())$ (line 6) process the SPARQL query described in Listing 3 to obtain the objects' textual description. After computing the score of the feature f, the function $updateScore(p, e.f)$ updates the score of p if the textual score of f is higher than the current score of p (line 8).

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

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

Listing 4: SPARQL query to obtain the objects of interest to process SKPQ-LD

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

6 Experimental Evaluation

In this section, we present our methodologies and the results obtained during the experimental evaluation. In addition, we discuss the dataset and the methodologies employed to analyze the proposed algorithm. The experiments were performed in two ways, each with a unique methodology. In the first experiment, the users' ratings from Google Maps were extracted to evaluate the queries result. In the second experiment, the users' ratings were extracted from TripAdvisor[3].

6.1 Datasets

In this work, we used three datasets to process the SKPQ. The OpenStreetMap (http://www.osm.org) dataset was used to process SKPQ and, DBpedia and LinkedGeoData were used to process SKPQ-LD. Additionally, two publicly available datasets were used to evaluate the obtained query results:: the Google Maps dataset and the OpinRank dataset.

Extracts are pieces of OpenStreetMap data pruned at the region of individual continents, countries, or metropolitan areas. Mapzen[4] maintains updated extracts for many cities. In this work, we used Mapzen to obtain OpenStreetMap data from Dubai. We process this dataset to extract only spatio-textual objects. The set of objects of interest P is composed by spatial objects whose the category in the OpenStreetMap is hotel, while the set of features F is composed by the other spatio-textual objects. The OpenStreetMap extract representing Dubai generated 162 objects of interest, 2243 features, 1906 unique terms and 12256 terms in total.

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

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

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

principles. To process SKPQ-LD we used SPARQL at LinkedGeoData to obtain a set of objects P equivalent to the one obtained from Mapzen, as illustrated by Listing 4. This SPARQL query returns a list of objects with the same name as the one stored at Mapzen, but different spatial coordinates (i.e. there are several places called "McDonald's" in Dubai, but at different spatial coordinates). Then, we selected only the object with the same name and the same spatial coordinate as the one selected as p object at Mapzen. Additionally, we used the LinkedGeoData endpoint to access feature's textual description. The textual description obtained from LinkedGeoData is composed by *rdf:type* and *rdfs:label* predicates.

In order to enrich the object's textual description from LinkedGeoData, we used the data obtained from DBpedia. The DBpedia project has derived its data corpus from the Wikipedia encyclopedia, a large collaborative encyclopedia. When a feature has the same *rdfs:label* in DBpedia and in LinkedGeoData, we concatenate the text obtained in both datasets. The textual description $f.D$ obtained from DBpedia is composed by *rdfs:comment* and *dbo:abstract* predicates. As an example, the Hotel Danieli from Venice is described as "(tourism) (hotel) Danieli" in OpenStreetMap. While in DBpedia, the same hotel is described as "Hotel Danieli, formerly Palazzo Dandolo, is a five-star palatial hotel in Venice, Italy. (..)"[5]. The hotel description in DBpedia is much wider than the OpenStreetMap description, with 58 more words.

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

6.1.1 Dataset for Experiment 1

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

6.1.2 Dataset for Experiment 2

The OpinRank dataset [Ganesan and Zhai 2011] contains hotel reviews and aspect ratings. There are 5 aspects ratings related to hotels: *cleanliness*, *value*,

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

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

Table 1: Example of information available in OpinRank dataset related to the query “great location”

service, *location* and *room*. The aspect ratings values are on a scale of 1-5. Ganesan and Zhai [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, so they reflect a natural user query. For example, the query “great location” is related to the aspect rating *location*. Given the query, the dataset lists the aspect rating value of each hotel as described in Table 1. 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 1.

6.2 Methodology

The DBpedia and LinkedGeoData were accessed through the local repository, or by the Snorql endpoint, as explained in Subsection 6.1. All experiments were executed in the same computer with an Intel Processor of 1.8 GHz (model i3-3217U) and 8 GB of RAM memory. For processing the SKPQ we made use of OpenStreetMap dataset, while for SKPQ-LD we used DBpedia dataset merged with OpenStreetMap dataset using SPARQL queries as discussed in Section 5.

The experiments were employed with 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 were executed twenty times using one unique query keyword each time. Half of the keywords are the most frequent terms in the dataset, the other half were randomly obtained. The query results were evaluated using NDCG. The list of frequent terms was obtained from S2I[6] and random queries keywords were obtained without repetition from a set of 1906 terms extracted from the OpenStreetMap dataset. “chili” and “sunset” are examples of random keywords used in this work. We used the object rate obtained from Google Places API to determine the ideal ranking.

[6] Implementation available at XXL Library

In Experiment 2, we apply the second methodology, where SKPQ and SKPQ-LD were 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 Subsection 6.1. We use the queries related to each aspect as query keywords and evaluate the query result obtained by SKPQ and SKPQ-LD. We ordered the query result by the aspect rating value of each hotel to determine the ideal ranking.

6.3 Metrics

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

The NDCG is widely used in Information Retrieval, measuring the quality of the ranking produced by a system [Baltrunas et al. 2010, Järvelin and 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 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 1 and 2, respectively. We denote top-k items by $P_k = \{p_1, p_2, ..., p_k\}$, where the items are ranked by the SKPQ and SKPQ-LD; and we denote rel_i as the relevance value of the item at position i. DCG@k is defined as

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

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

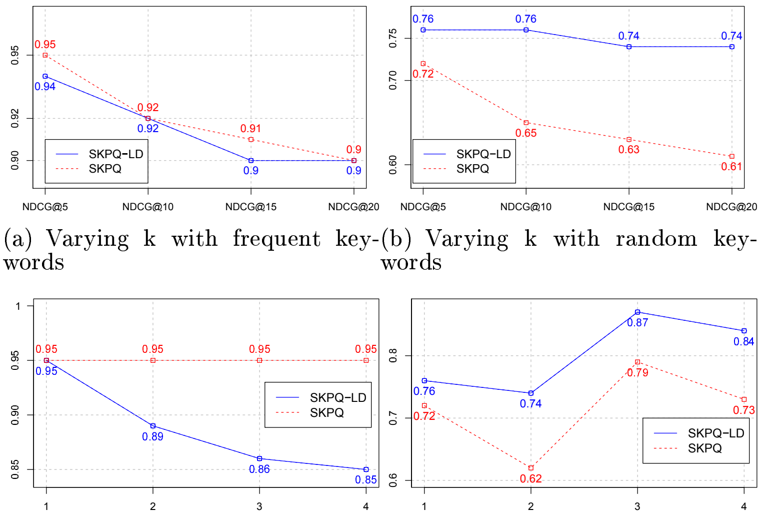
$$IDCG = max(DCG@k) \tag{2}$$

NDCG@k is calculated as

$$NDCG@k = \frac{DCG@k}{IDCG} \tag{3}$$

6.4 Experiment 1: Evaluating Query Results

To understand the ranking quality of both SKPQ and SKPQ-LD, we compared the NDCG values obtained when using random keywords and frequent keywords. Figure 2 reports the arithmetic mean of NDCG@k (k=5, 10, 15, 20) that are generated by the queries with different keywords. The arithmetic mean values are reported on the vertical axis. Figures 2(a) and 2(b) illustrate that SKPQ-LD improves the ranking quality when using random keywords, otherwise the quality is roughly the same.



(a) Varying k with frequent keywords

(b) Varying k with random keywords

(c) Varying keywords using frequent terms

(d) Varying keywords using random terms

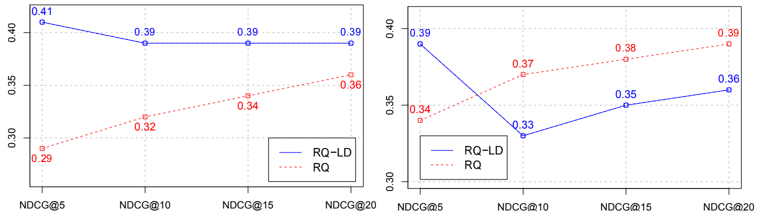
Figure 2: Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size (k)

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

Figures 2(c) and 2(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 2(c) used the 10 most frequent terms in the dataset as query keywords. To build query keywords with 2 terms or more, we combined these terms with each other without repetition.

As it can be seen in Figure 2(c), even after adding three more keywords, the results obtained in SKPQ does not change. On the other hand, SKPQ-LD is more influenced by the increase in the number of query keywords. As observed in Figure 2, 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 2(c) slowly decreases as the number of keywords grows. In addition, we noticed that the SKPQ results had few, or none, changes when the number of keywords was increased. For example, the query result for the keywords “parking cafe” was equal to the query results obtained with “bank parking cafe” and “parking supermarket cafe bank”. The textual score of each object presented had changed, but there was no difference on the rank order, resulting in similar NDCG values. The SKPQ lacks a result variability because of the poor textual description of its objects. SKPQ-LD obtained lower NDCG values but did present different results to each query keyword.

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



(a) Using frequent keywords (b) Using random keywords

Figure 3: Results obtained with RQ and RQ-LD

It can be seen in Figure 3 that our approach improved RQ result set when using frequent keywords instead of random keywords. The RQ looks for all k objects in a small spatial area (radius = 200 m) while SKPQ looks for objects in the neighborhood of many objects of interest. Each object neighborhood has the same size of all the spatial area visited by RQ (200 m). This contrast results in a more challenging effort to build a quality rank for the given area because there are fewer objects to verify. This can be verified observing the much lower

NDCG values obtained with RQ. While SKPQ obtained 0.61 in its worst case, RQ obtained 0.41 as its best case. The amount of objects to verify is the main reason for the lower NDCGs values depicted in Figure 3 than the ones in Figure 2.

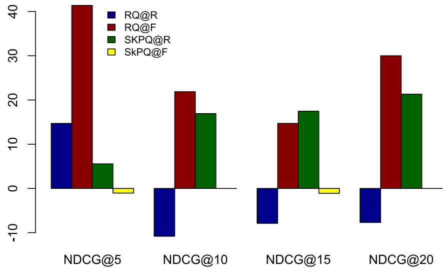


Figure 4: Relative NDCG improvements

Figure 4 illustrates the relative NDCG improvement (as described in [Song et al. 2016]) of the proposed approach e_{pro} over respective baseline model e_{other}, further measured as

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

Figure 4 reports the relative NDCG improvement values on the vertical axis. The proposed approach demonstrated different degrees of improvement in different scenarios. It improved SKPQ relative NDCG in 20% when using random keywords (SKPQ@R - NDCG@20) and 40% when RQ used frequent keywords (RQ@F - NDCG@5).

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

6.5 Experiment 2: Evaluating feature selection

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

Given the query keywords, the SKPQ returns a list of objects of interest whose are near to features relevant to the given query keywords. We desire that

SKPQ returns objects whose features have a high aspect rating value. This way, the SKPQ would be selecting good features according to users of TripAdvisor. If there is no relevant feature near an object of interest, the SKPQ query result is empty.

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

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

We show examples of textual queries that we could extract rating values, and those we could not, to illustrate this scenario. The queries "nice staff" and "good value" are examples of queries that did not return any relevant objects to the user. The objects textual description in SKPQ and SKPQ-LD was not able to describe these aspects of the hotels. However, the queries "great location", "clean place" and "cozy rooms" returned objects when using SKPQ-LD. Figure 5 reports the NDCG values of the query results obtained with these query keywords.

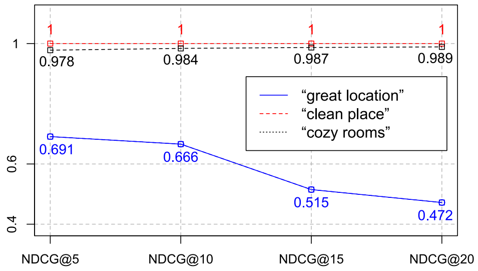


Figure 5: SKPQ-LD evaluation using OpinRank

With the enhancing of objects' textual description, SKPQ-LD was able to select more objects that satisfy the user need than SKPQ. Accordingly to the obtained NDCG values in Figure 5, SKPQ-LD selected features of good quality. Since the query results have high aspect rating values, we can assume that SKPQ-LD was able to find good objects to the user. For the query "clean place" for example, SKPQ-LD was able to find features that are evaluated by real users as a clean hotel.

The OpinRank dataset contains other queries created by the combination of the queries illustrated in Figure 5 plus the queries "nice staff" and "good value". Nevertheless, the combination of these queries lead to results very similar to the ones at Figure 5. In this experiment, the SKPQ-LD demonstrated that the textual description improvement enhances the query capabilities, enabling it to find more objects. Without the textual description improvement, the SKPQ was unable to find any relevant objects to the presented queries.

7 Discussion

In this section, we discuss how the textual description enrichment affects the results obtained in ours two experiments and the limitations of our approach. We ran the SKPQ varying query keywords and extend our analysis to understand the difference between textual descriptions from DBpedia and OpenStreetMap. Table 2 is an experiment result using hotels from Venice as objects of interest (P) and "church" as a query keyword. The first column of Table 2 presents the object of interest textual description, the second column has the object of interest score using traditional SKPQ, and the third column presents the object of interest score using SKPQ-LD. In order to find features that satisfy the spatial selection criteria using LOD, the *geo:geometry* property has to exist in LOD object. For this reason, the objects of interest "Palazzo Ferro Fini" and "Splendid Venice" has no score.

Some cities (e.g. Venice) has few spatial objects of interest represented at DBpedia. This database contains only 5 hotels in Venice against 488 registered in OpenStreetMap. Despite the great number of objects, the textual description in OpenStreetMap is poor. While a typical textual description in DBpedia has around 60 terms, the textual description for the same object has only 2 terms in OpenStreetMap. The poor textual description leads the SKPQ to misjudge the evaluation of some objects of interest, as can be seen in Table 2. Given the query keyword "church", objects "Hotel Cipriani" and "Hotel Danieli" have features in their spatial neighborhood that are textual relevant to the query keyword, but traditional SKPQ fails to identify them because of poor textual description. SKPQ-LD did find these objects and was able to evaluate "Hotel Cipriani" and "Hotel Danieli". For the same reason, SKPQ did not find relevant objects in the Experiment 2 described in Subsection 6.5.

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

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

Table 2: Score of Object p in Traditional SKPQ Compared With the Score Generated by SKPQ-LD, using hotels from Venice

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

Table 3: Score of Object p in Traditional SKPQ Compared With the Score Generated by SKPQ-LD, using hotels from São Paulo

"church" as a query keyword was presented in Table 3. The column names in Table 3 have the same meaning as the column names in Table 2. This time we have no problem to find the *geo:geometry* property but SKPQ still has problems to evaluate objects. SKPQ still returns more objects with score zero than SKPQ-LD. These results endorse the improvement obtained by our approach when using random query keywords since "church" is a random query keyword.

As illustrated in Table 3, the SKPQ score of the object "San Michel Hotel" is higher than its SKPQ-LD score. When the query keyword has only one term, the textual score takes into account only the length of the document (number of terms) and the term impact. Using traditional SKPQ, we expect a higher object p score than the one computed by SKPQ-LD. The score in traditional SKPQ is higher than SKPQ-LD because the document length is shorter, therefore the term impact in this document is more evident.

7.1 Limitations and Points of Improvements

Despite the obtained results look promising, our approach has some limitations. First, although the LOD cloud increases every day, textual descriptions may not always be available with expected quality. This may eventually penalize the query results when using LOD. For instance, the hotel "Splendid Venice" (presented at Section 7) does not have the *geo:geometry* property hindering the textual description access by spatial queries.

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

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

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

8 Conclusion

In this paper, we proposed an enhancement to Top-k Spatial Keyword Preference Query. This enhancement uses LOD to improve the textual description of features. We have presented how to obtain the textual description using SPARQL and an algorithm to process this query with data available at LOD cloud. Results from our experiments show that a richer textual description (obtained from LOD datasets) can contribute to enhancing the SKPQ query result.

A larger textual description was employed to present results for the user in situations where traditional SKPQ could not. In the first experiment conducted, evaluating the query results with Google Maps dataset, we observed that our method can perform 20% better than the traditional approach. This takes place because all objects had a wider textual description when processing the query using our approach. Also, in Experiment 2 we observed that using real queries obtained from OpinRank dataset, the SKPQ was unable to find features without using our approach. In addition to finding these features, we observed that the selected features have good quality according to TripAdvisor users.

In future works, we aim to create an evaluation model using expertise judgments. This will give a more precise evaluation about the SKPQ-LD. We also plan to extend the algorithm, enabling richer textual descriptions. Moreover, we also have the intention to evaluate the response time and I/O of the SKPQ-LD. These measures will be useful to analyze the impact of LOD on query processing performance.

References

[Baltrunas et al. 2010] Baltrunas, L., Makcinskas, T. and Ricci, F.: "Group recommendations with rank aggregation and collaborative filtering"; Proc. of the fourth ACM conference on Recommender systems, ACM, (2010), 119-126.

[Becker and Bizer 2009] Becker, C. and Bizer, C.: "Exploring the geospatial semantic web with dbpedia mobile"; Web Sem.: Sci., Serv. and Ag. on the World Wide Web, Elsevier, 7, 4 (2009), 278-286.

[Becker and Bizer 2009] Bizer, C., Heath, T. and Berners-Lee, T.: "Linked Data - the story so far"; Intern. Journal on Sem. Web and Info. Sys., 5, 3 (2009), 1-22. https://eprints.soton.ac.uk/271285/.

[Berners-Lee et al. 2005] Berners-Lee, T., Fielding, R. and Masinter, L.: "Rfc 3986"; Uniform Resource Identifier (URI): Generic Syntax, (2005).

[Berners-Lee 2011] Berners-Lee, T.: "Design issues: Linked data (2006)" (2011). http://www.w3.org/DesignIssues/LinkedData.html.

[Bizer et al. 2012] Bizer, C., Boncz, P., Brodie, M. L. and Erling, O.: "The meaningful use of big data: four perspectives–four challenges"; ACM Sigmod Record, ACM, 40, 4 (2012), 56-60.

[Braun et al. 2010] Braun, M., Scherp, A. and Staab, S.: "Collaborative creation of semantic points of interest as linked data on the mobile phone"; Arbeitsberichte aus dem Fachbereich Informatik, Koblenz (2010).

[Cao et al. 2012] Cao, X., Chen, L., Cong, G., Jensen, C. S., Qu, Q.: "Spatial keyword querying"; Intern. Conf. on Conc. Modeling, Springer, Berlin (Oct 2012), 16-29.

[Cong et al. 2009] Cong, G., Jensen, C. S. and Wu, D.: "Efficient retrieval of the top-k most relevant spatial web objects"; Proc. of VLDB, VLDB End., (2009), 337-348.

[de Almeida and Rocha-Junior 2016] de Almeida, J. P. D. and Rocha-Junior, J. B.: "Top-k spatial keyword preference query"; JIDM, 6, 3 (2016), 162-178.

[Fernández-Tobías et al. 2011] Fernández-Tobías, I., Cantador, I., Kaminskas, M. and Rici, F.: "A generic semantic-based framework for cross-domain recommendation"; Proc. of the 2nd Intern. Workshop on Information Heterogeneity and Fusion in Rec. Sys., ACM, (Oct 2011), 25-32.

[Fielding et al. 1999] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T.: "Hypertext transfer protocol–HTTP/1.1". (1999).

[Ganesan and Zhai 2011] Ganesan, K. and Zhai, C.: "Opinion-based entity ranking"; Information retrieval, Springer, 15, 2 (2012), 116-150.

[Gracia et al. 2012] Gracia, J., Montiel-Ponsoda, E., Cimiano, P., Gómez-Pérez, A., Buitelaar, P. and Mc-Crae, J.: "Challenges for the multilingual web of data"; Web Sem.: Sci., Serv. and Ag. on the World Wide Web, Elsevier, (2012), 63-71.

[Hegde et al. 2011] Hegde, V., Reynolds, V., Parreira, J. X. and Hauswirth, M.: "Utililising Linked Data for Personalized Recommendation of POI's"; Intern. AR Stand. Meet., Barcelona (2011).

[Järvelin and Kekäläinen 2002] Järvelin, K. and Kekäläinen, J.: "Cumulated gain-based evaluation of IR techniques"; Trans. on Info. Sys., ACM, 20, 4 (2002), 422-446.

[Karam and Melchiori 2013] Karam, R. and Melchiori, M.: "Improving geo-spatial linked data with the wisdom of the crowds"; Proc. of the joint EDBT/ICDT, ACM, (Mar 2013), 68-74.

[Papadias et al. 2001] Papadias, D., Kalnis, P., Zhang, J. and Tao, Y.: "Efficient OLAP operations in spatial data warehouses"; Intern. Sym. on Spatial and Temporal Data., Springer, (2001), 443-459.

[Perry and Herring 2012] Perry, M. and Herring, J.: "OGC GeoSPARQL-A geographic query language for RDF data"; OGC implementation standard, (2012).

[Rocha-Junior et al. 2011] Rocha-Junior, J. B., Gkorgkas, O., Jonassen, S. and Nørvåg, K.: "Efficient processing of top-k spatial keyword queries"; Intern. Sym. on Spatial and Temporal Data., Springer, Berlin (2011), 205-222.

[Salton and Buckley 1988] Salton, G. and Buckley, C.: "Term-weighting approaches in automatic text retrieval"; Info. proc. & manag., Elsevier, 24, 5 (1988), 513-523.

[Seo et al. 2018] Seo, Y.-D., Kim, Y.-G., Lee, E., Seol, K.-S. and Baik, D.-K.: "An enhanced aggregation method considering deviations for a group recommendation"; Expert Systems with Applications, Elsevier, (2018), 299-312.

[Song et al. 2016] Song, H., Xu, Y., Min, H., Wu, Q., Wei, W., Weng, J., Han, X., Yang, Q., Shi, J., Gu, J. et al.: "Individual Judgments Versus Consensus: Estimating Query-URL Relevance"; ACM Trans. on the Web, ACM, 10, 1 (2016).

[Thompson et al. 2017] Thompson, Paul and Nawaz, Raheel and McNaught, John and Ananiadou, Sophia: "Enriching news events with meta-knowledge information"; Language Resources and Evaluation, Springer, 51, 2 (2017).

[Wang et al. 2015] Wang, J., Yu, C. T., Yu, P. S., Liu, B. and Meng, W.: "Diversionary comments under blog posts"; ACM Trans. on the Web, ACM, 9, 4 (2015), 443-459.

[Zarrinkalam and Kahani 2012] Zarrinkalam, F. and Kahani, M.: "A multi-criteria hybrid citation recommendation system based on linked data"; Computer and Knowledge Engineering, IEEE, (2012), 283-288.

[Zobel and Moffat 2006] Zobel, J. and Moffat, A.: "Inverted files for text search engines"; ACM comp. surveys, ACM, 38, 2 (2006).