

Improving the Spatial Keyword Preference Query with Linked Open Data

João Paulo Dias de Almeida
Federal University of Bahia
Av. Adhemar de Barros, Ondina
Salvador, Bahia 40170-115
joao.dias@ufba.br

Frederico Araújo Durão
Federal University of Bahia
Av. Adhemar de Barros, Ondina
Salvador, Bahia 40170-115
fdurao@ufba.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 and two experimental evaluations of the proposed approach, 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.

KEYWORDS

Spatial data, Query evaluation, Query processing, Linked Open Data

1 INTRODUCTION

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". These queries evaluate an object as relevant for the user when it's textual description shares words in common with the query keywords [3, 4]. 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. It is possible that an object is relevant for the user but its description does not contain any query keyword. One possible solution for this problem is offering a wider textual description for the objects.

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 [6, 9] and Information Retrieval [1, 11]. 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) [5]. 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 a novel semantic model for enhancing the SKPQ and 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 describes the SPARQL queries employed, then Sections 4 and 5 present the experimental evaluations and the discussion about these evaluations. Finally, Section 6 presents the conclusions and future work.

2 MOTIVATING SCENARIO

The Top-k Spatial Keyword Preference Query (SKPQ) is a preference query that uses query keywords to describe the user preference [5]. The SKPQ searches for spatial objects of user's interest based on spatio-textual objects¹ 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.

Suppose a SKPQ with query keywords "oriental food". Considering Figure 1, this query does not return any objects. Neither

In: XVIII Workshop de Teses de Dissertações (WTD 2018), Salvador, Brasil. Anais do XXIV Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pósteres. Porto Alegre: Sociedade Brasileira de Computação, 2018.
© 2018 SBC – Sociedade Brasileira de Computação.
ISBN 978-85-7669-435-9.

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

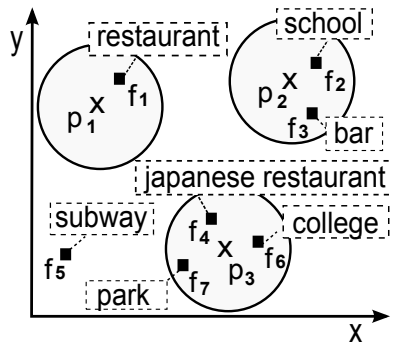


Figure 1: Spatial objects of interest (p) and features (f) associated with their textual descriptions.

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 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://linkedgeo.org/sparql>. Listing 1 introduces a SPARQL query to obtain features within 200 m from an object of interest. In Listing 1, *objectURI* is a URI to an object of interest.

The predicate *geo:geometry* is defined at Geo-SPARQL [12], an ontology that represents features and geometries. In Listing 1, 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 1.

4 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 approach. 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

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

Listing 1: SPARQL query to find features that satisfies the spatial selection criteria.

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

Listing 2: SPARQL query to obtain the objects of interest to process SKPQ-LD.

result. In the second experiment, the users’ ratings were extracted from TripAdvisor².

4.1 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. Mapzen³ 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 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 2. 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.

²<https://www.tripadvisor.com.br/>

³<https://mapzen.com/data/metro-extracts/>

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”.

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.

4.1.1 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.

4.1.2 Dataset for Experiment 2. The OpinRank dataset [7] 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 [7] 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.

4.2 Methodology

The DBpedia and LinkedGeoData were accessed through the local repository, or by the Snorql endpoint, as explained in Subsection 4.1. 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.

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 S2I⁴ and random queries keywords were obtained without repetition from a set of 1906 terms extracted from the OpenStreetMap dataset. “chili” and “sunset” are examples of random keywords used in this work. We used the object rate obtained from Google Places API to determine the ideal ranking.

In Experiment 2, 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 4.1. We use the queries related to each aspect as query keywords and evaluate the query result obtained by SKPQ and SKPQ-LD. We ordered the query result by the aspect rating value of each hotel to determine the ideal ranking.

The metrics employed in all experiments were Discount Cumulative Gain (DCG), Normalized Discount Cumulative Gain (NDCG) and Mean Average Precision (MAP). The NDCG is widely used in Information Retrieval, measuring the quality of the ranking produced by a system [10]. 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 at position k (DCG@k) and ideal DCG (IDCG) are calculated based on Equation 1 and 2, respectively. We denote top-k items by $P_k = \{p_1, p_2, \dots, p_k\}$, 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 .

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

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

$$IDCG = \max(DCG@k) \quad (2)$$

So, NDCG@k is calculated as

$$NDCG@k = \frac{DCG@k}{IDCG} \quad (3)$$

4.3 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 3 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 3(a) illustrate that SKPQ-LD improves the ranking quality when using random keywords, otherwise the quality is roughly the same.

It is noticeable that we obtain better results with SKPQ using frequent keywords. Since the keyword is present in many objects, there is no problem to SKPQ identify the object that has textual relevance to the query keyword. In this scenario, the objects in

⁴Implementation available at XXL Library

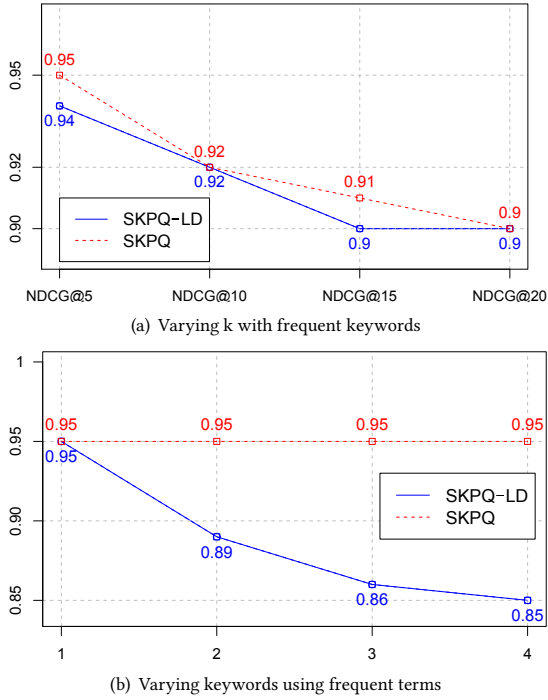


Figure 2: Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size (k)

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(b) and 3(b) 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(b) 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(b), 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 3, 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(b) 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

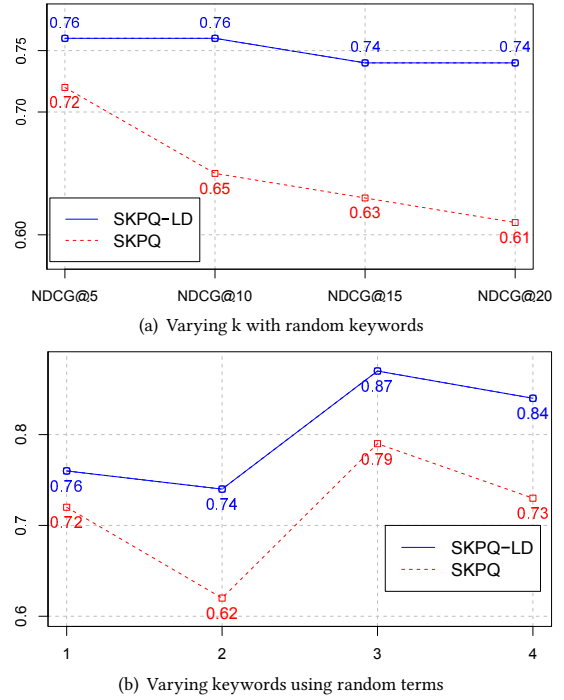


Figure 3: Results obtained by SKPQ and SKPQ-LD varying the keywords and the query result size (k)

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) [3] 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.

It can be seen in Figure 4 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 4 than the ones in Figure 3.

Figure 5 illustrates the relative NDCG improvement (as described in [13]) of the proposed approach e_{pro} over respective baseline model e_{other} , further measured as

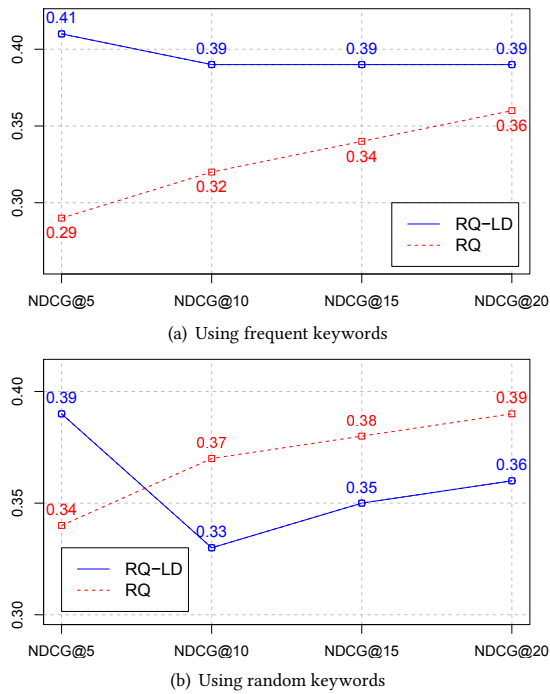


Figure 4: Results obtained with RQ and RQ-LD

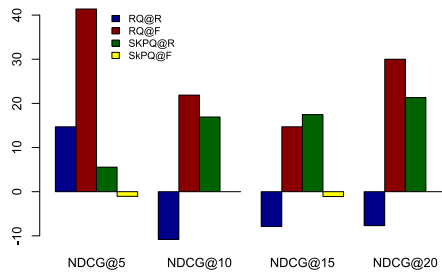


Figure 5: Relative NDCG improvements

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

Figure 5 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.

4.4 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 and are relevant to the given query. 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 6 reports the NDCG values of the query results obtained with these query keywords.

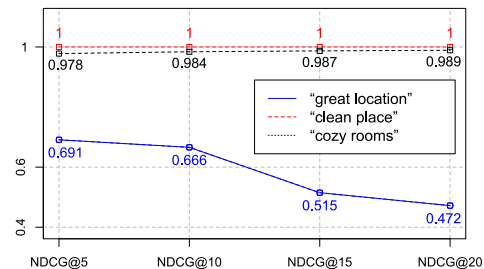


Figure 6: SKPQ-LD evaluation using OpinRank

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 6, 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 6 plus the queries “nice staff” and “good value”. Nevertheless, the combination of these queries lead to results very similar to the ones at Figure 6. In this experiment, the SKPQ-LD demonstrated that the textual description improvement enhances the query capabilities, enabling it to find more objects. Without the textual description improvement, the SKPQ was unable to find any relevant objects to the presented queries.

5 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.

Zarrinkalam and Kahani [14] 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 [8] and [2]. 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 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. Results from our experiments show that a richer textual description 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 at creating 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.

Acknowledgments. We thank FAPESB for the financial support of this research.

REFERENCES

- [1] Christian Becker and Christian Bizer. 2009. Exploring the geospatial semantic web with dbpedia mobile. *Web Semantics: Science, Services and Agents on the World Wide Web* (2009), 278–286.
- [2] Christian Bizer, Peter Boncz, Michael L Brodie, and Orri Erling. 2012. The meaningful use of big data: four perspectives—four challenges. *ACM SIGMOD Record* 40, 4 (2012), 56–60.
- [3] Xin Cao, Lisi Chen, Gao Cong, Christian S. Jensen, Qiang Qu, Anders Skovsgaard, Dingming Wu, and Man Lung Yiu. 2012. Spatial keyword querying. In *ER*. Springer, 16–29.
- [4] Gao Cong, Christian S. Jensen, and Dingming Wu. 2009. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB* 2, 1, 337–348.
- [5] João Paulo Dias de Almeida and João B Rocha-Junior. 2016. Top-k Spatial Keyword Preference Query. *Journal of Information and Data Management* (2016), 162.
- [6] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskis, and Francesco Ricci. 2011. A generic semantic-based framework for cross-domain recommendation. In *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*. ACM, 25–32.
- [7] Kavita Ganesan and ChengXiang Zhai. 2011. Opinion-Based Entity Ranking. *Information Retrieval* (2011).
- [8] Jorge Gracia, Elena Montiel-Ponsoda, Philipp Cimiano, Asunción Gómez-Pérez, Paul Buitelaar, and John McCrae. 2012. Challenges for the multilingual web of data. *Web Semantics: Science, Services and Agents on the World Wide Web* 11 (2012), 63–71.
- [9] Vinod Hegde, Vinny Reynolds, Josiane Xavier Parreira, and Manfred Hauswirth. 2011. Utilising Linked Data for Personalized Recommendation of POI’s. In *International AR Standards Meeting, Barcelona, Spain*.
- [10] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
- [11] Roula Karam and Michele Melchiori. 2013. Improving geo-spatial linked data with the wisdom of the crowds. In *Proceedings of the joint EDBT/ICDT 2013 workshops*. ACM, 68–74.
- [12] Matthew Perry and John Herring. 2012. OGC GeoSPARQL-A geographic query language for RDF data. *OGC Implementation Standard*. Sept (2012).
- [13] Hengjie Song, Yonghui Xu, Huaqing Min, Qingyao Wu, Wei Wei, Jianshu Weng, Xiaogang Han, Qiang Yang, Jialiang Shi, Jiaqian Gu, et al. 2016. Individual Judgments Versus Consensus: Estimating Query-URL Relevance. *ACM Transactions on the Web (TWEB)* 10, 1 (2016), 3.
- [14] Fattane Zarrinkalam and Mohsen Kahani. 2012. A multi-criteria hybrid citation recommendation system based on linked data. In *Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on*. IEEE, 283–288.