Top-K Diversification for Path Queries in Knowledge Graphs

Christian Aebeloe[1], Vinay Setty[1,2], Gabriela Montoya[1], and Katja Hose[1]

[1] Aalborg University, Denmark
{caebel,gmontoya,vinay,khose}@cs.aau.dk
[2] University of Stavanger, Norway
vsetty@acm.org

Abstract. To explore the relationships between entities in RDF graphs, property path queries were introduced in SPARQL 1.1. However, existing RDF engines return only reachability of the entities ignoring the intermediate nodes in the path. If the paths are output, they are too many, which makes it difficult for users to find the most relevant paths. To address this issue, we propose a generalized top-k ranking technique that balances the trade-off between relevance and diversity. We propose a shortest path based relevance scoring in combination with several path similarity measures for diversification. With preliminary experiments and examples, we show that our diversification strategies provide more informative paths compared to shortest path based ranking.

1 Introduction

Knowledge Graphs (KGs) have become a popular way to represent and query world knowledge. KGs such as YAGO [7] and DBPedia [3] are extensively used for tasks such as question answering, knowledge exploration, and reasoning. RDF[3] and SPARQL[4] are standards for representing and querying KGs, recommended by the W3C and widely adopted by the semantic web community.

In SPARQL 1.1[5], property path queries were introduced to check only the transitive reachability of entities via paths of specific properties without the actual paths connecting the entities. For example, a property path query with the clause "`Alan_Turing linksTo* Princeton_University`" checks if `Princeton_University` is transitively reachable from `Alan_Turning` via any number of `linksTo` property relationships but omits the intermediate entities in the path. To address this issue, we have proposed an additional operator → that returns the full paths rather than only performing reachability checks [1]. Since enumerating all the paths may be overwhelming, we need ranking techniques to select top-k most informative paths to the users.

Related work on ranking and diversifying property path query results is currently very limited as supporting reachability checks already is a very challenging task. Some recent works rank the paths between entities based on their lengths and frequency of resources [5,9]. However, they only support queries involving a single property rather

[3] https://www.w3.org/RDF/

[4] https://www.w3.org/TR/rdf-sparql-query/

[5] https://www.w3.org/TR/sparql11-query/

than involving a diverse range of properties occurring in the KG. Therefore, we need a general top-k ranking technique supporting arbitrary property path queries.

Selecting paths based on top-k shortest path ranking is a good strategy in general, but shortest paths may have several redundant resources (entities and properties). Therefore, it is also essential to apply *diversification* techniques to avoid redundancy, while still minimizing the path lengths. However, existing techniques do not balance the trade-off between path lengths and diversity. The diversification of paths using the Jaccard Similarity measure has been proposed before [9] but they treat paths as sets and ignore the order of entities. To the best of our knowledge, there are no diversification measures considering the semantics of paths.

To address these issues, our contributions in this paper are: (1) we formulate a generalized top-k ranking technique for property paths that balances the trade-off between path lengths and diversity. (2) we then explore several path similarity measures for diversification and propose a Levenshtein Similarity measure for respecting path semantics. (3) we perform preliminary evaluations and compare the effectiveness of various diversification strategies using an evaluation metric based on the Normalized Discounted Cumulative Gain (NDCG) [6].

2 Diversification Metrics for Property Paths

Given a property path query Q and a set of all relevant paths R, our goal is to select $S \subseteq R$ to maximize the objective given below:

$$DivP(R) = \arg\max_{S \subseteq R} \sum_{P_i \in S} (1-\lambda) \cdot Rel(P_i, R) - \lambda \cdot Sim(P_i, R) \quad s.t. |S| = k \tag{1}$$

Where, $Rel(P_i, R)$ is a function quantifying the relevance of a path P_i to a query Q with result set R. Each path P contains a set of entities and property edges connecting them which we call "resources" and each path has a path length represented as $|P|$ which is the total number of resources in the path.

To avoid paths with redundant resources, the objective function in Equation 1, aims for balancing the relevance (Rel) and the similarity (Sim) among paths in S. The trade-off between these two scores are balanced by a user-specified diversification parameter $0 \leq \lambda \leq 1$. If $\lambda = 0$, we rank the paths purely according to their lengths. On the other hand, if $\lambda = 1$, the k most dissimilar paths according to some similarity measure irrespective of their path lengths are chosen. Computing a solution for $DivP(R)$ is known to be NP-hard [4]. However, since the objective follows the submodularity property, there are known greedy heuristics for solving it approximately [1,8].

In this paper, we quantify the relevance relative to the shortest path in the result set. A path is more relevant if its path length is closer to the shortest path's length. We compute the normalized relevance score as $Rel(P_i, R) = (\min_{P_j \in R} |P_j|)/|P_i|$. We can also use other scoring models such as weighted path lengths and informativeness measures [9]. In this paper, we explore several Sim functions that we can use for diversification.

Jaccard Similarity: The simplest way to quantify the overlap in paths is to treat them as sets and compute the Jaccard similarity value. Jaccard similarity has been used for of path queries before [1,9]. Jaccard similarity is defined as:

$$Sim_J(P_i, R) = \max_{P_j \in S, P_j \neq Pi} \frac{|P_i \cap P_j|}{|P_i \cup P_j|} \tag{2}$$

Since Jaccard similarity captures the overlap in paths by treating them as sets, it ignores the order of resources in the paths. Intuitively, Sim_J cannot distinguish between two paths with identical resources but connected in a different order. To address this issue, we need a similarity measure which preserves the order of sequences.

Levenshtein Similarity: To quantify the path sequence similarity, we adapt the Levenshtein distance[6] (LD), which is generally used to quantify the distance between two strings as the minimal number of insertions, deletions, and substitutions of one character for another that will transform one string into the other. For diversification of property paths, in Equation 1, we normalize the similarity value computed from LD as follows:

$$Sim_L(P_i, R) = \max_{P_j \in S, P_j \neq Pi} 1 - \frac{LD(P_i, P_j)}{max(|P_i|, |P_j|)} \tag{3}$$

3 Evaluation

For evaluation we use the YAGO knowledge base, which contains $1+$ Billion triples [7]. To evaluate the effectiveness of the aforementioned diversification metrics, we define an evaluation metric similar to the one by Arnaou et al. [2] based on NDCG [6]. Evaluations are performed on sets of top-k paths of R, sorted according to equation 1 in descending order.We define the evaluation metric as $EvalP(R) = DCG(R)/IDCG(R)$, where $DCG(R)$ is defined as below:

$$DCG(R) = Rel(P_1, R) + Nov_1 + \sum_{i=2}^{k} \left(\frac{Rel(P_i, R)}{log_2(i)} + Nov_i \right) \quad s.t. P_i \in S \tag{4}$$

$IDCG(R)$ is the ideal DCG obtained by sorting $Rel(P_i, R)$. The novelty is defined as $Nov_i = \frac{\#unseen}{\#resources}$, where $\#unseen$ is the number of unseen resources at rank i.

λ	0	0.2	0.4	0.6	0.8	1
Jaccard	1	0.996	0.965	0.908	0.866	0.806
Levenshtein	1	0.997	0.978	0.951	0.91	0.889
Probabilistic	1	1	0.999	0.938	0.887	0.84

Table 1. Average NDCG-values for top-100 paths.

We perform evaluations on 10 randomly chosen queries. In Table 1, we measure the NDCG varying the λ-values and fix $k = 100$. Note that, both diversification metrics have similar NDCG-values for most λ-values. However, with low λ-values, Levenshtein has an advantage over Jaccard. This is because, with $\lambda = 0$, only relevance is considered and as the λ-value grows, more emphasis is put on the diversification metric. And since Levenshtein puts emphasis on differences in path sequences rather than resources.

[6] For definition and formula see https://en.wikipedia.org/wiki/Levenshtein_distance

To illustrate this, consider the path query:

```
PREFIX yago: <http://yago-knowledge.org/resource/>
SELECT DISTINCT * WHERE {
  yago:Alan_Turing ?p→[4] yago:Princeton_University
}
```

Listing 1.1. SPARQL query Q1 to retrieve top-4 paths between Alan_Turing and Princeton_University using → operator

```
1. Alan_Turing → graduatedFrom → Princeton_University
2. Alan_Turing → linksTo → Bertrand_Russell → linksTo → John_Dewey →
       influences → Cornel_West → linksTo → Princeton_University
3. Alan_Turing → hasAcademicAdvisor → Alonzo_Church → hasAcademicAdvisor →
       Oswald_Veblen → worksAt → Princeton_University
```

Listing 1.2. Top-3 results for Q1 using Jaccard and Levenshtein similarity are identical

The top-3 results for the query Q1 are the same for both similarity measures. However, Levenshtein has Alan_Turing → linksTo → Princeton_University at fourth place, which is not included at all in top-10 with Jaccard similarity, as it is contained in other paths. But according to Levenshtein, it is quite different from other paths.

4 Conclusion

In this paper, we addressed the issue of ranking paths in KGs. We proposed a generalized top-k diversification technique which is customizable with different relevance and similarity functions. To remedy the limitations of Jaccard similarity, we proposed the Levenshtein measure which respects the order of nodes in the paths. We conducted preliminary experiments using an NDCG-based metric to show that Levenshtein metric provides better diversification than Jaccard. In future, we will propose indexing and query optimizations for these diversification measures.

Acknowledgment This research was partially funded by the Danish Council for Independent Research (DFF) under grant agreement no. DFF-4093-00301 and Aalborg University's Talent Management Programme.

References

1. C. Aebeloe, G. Montoya, V. Setty, and K. Hose. Discovering Diversified Paths in Knowledge Bases. *PVLDB*, 2018. Demonstration. to appear.
2. H. Arnaout and S. Elbassuoni. Result Diversity for RDF Search. In *KDIR*, pages 249–256, 2016.
3. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*, pages 722–735. Springer, 2007.
4. J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98*, pages 335–336, 1998.
5. V. Fionda and G. Pirrò. Explaining and querying knowledge graphs by relatedness. *PVLDB*, 10(12):1913–1916, 2017.
6. K. Järvelin and J. Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.
7. F. Mahdisoltani, J. Biega, and F. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR'14*, 2014.

8. G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming*, 14(1):265–294, 1978.
9. G. Pirrò. Explaining and suggesting relatedness in knowledge graphs. In *ISWC*, pages 622–639, 2015.