Towards Multi-Facet Snippets for Dataset Search

Xiaxia Wang[1], Gong Cheng[1], and Evgeny Kharlamov[2,3]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, China
xxwang@smail.nju.edu.cn, gcheng@nju.edu.cn
[2] Department of Informatics, University of Oslo, Norway
evgeny.kharlamov@ifi.uio.no
[3] Bosch Center for Artificial Intelligence, Renningen, Germany
evgeny.kharlamov@de.bosch.com

Abstract. Due to a recent significant increase in the number of RDF datasets available on the Web, there is a pressing need in effective search techniques for finding the right data on demand. A promising approach is to present retrieved datasets as snippets that aim at concisely explaining to the user why this dataset fulfils their demand. Snippets in particular can illustrate the main content of the dataset and explain its relevance to the user's query. Computing optimal snippets is a non-trivial task and a number of approaches have emerged to address this problem. In this short paper, we report our ongoing work on snippets that address multiple facets of optimality. Based on our recently proposed evaluation metrics for dataset snippets, we formulate a weighted maximum coverage problem which directly optimizes three evaluation metrics. We solve the problem with a greedy algorithm, and our current implementation has outperformed four baseline methods.

1 Introduction

The open data movement brings increasingly many datasets to the Web, many of which are in the RDF format. Reusing these datasets is of great importance to researchers and developers. In order to enable the reuse there is a pressing need in effective search techniques for finding the *right* data on demand. A promising approach is to query for datasets with keywords as in Google Dataset Search [1] and to present each retrieved RDF dataset as a *snippet*, its small representative subset [2]. Dataset snippets aim at concisely explaining to the user *why* this dataset fulfils their demand and in particular can illustrate the main content of the dataset and explain its relevance to the user's query.

Computing optimal snippets is a non-trivial task and a number of approaches have emerged to address this or related problems [2–6]. In [7], we presented four metrics for evaluating the quality of a dataset snippet. In this short paper, we report our ongoing work on snippets that address multiple facets of optimality. In particular, in order to improve the quality of a snippet for dataset search, we

formulate the selection of RDF triples as a combinatorial optimization problem that directly optimizes three evaluation metrics proposed in [7]. A dataset snippet generated by our approach, which we refer to as **KSD**, is expected to have a good coverage of the query **K**eywords and the content of the dataset at both the **S**chema and the **D**ata level. We solve the problem with a greedy algorithm, and our evaluation demonstrates that KSD outperforms the baselines reported in [7] and that there is still a considerable room for quality improvement.

The remainder of this paper is structured as follows. Section 2 defines the problem and reviews the evaluation metrics proposed in [7]. Section 3 describes the implementation of KSD. Section 4 presents evaluation results. Section 5 concludes the paper with future work.

2 Preliminaries

2.1 Problem Statement

An RDF dataset is a set of RDF triples denoted by $T = \{t_1, t_2, \cdots, t_n\}$, where each $t_i = \langle t_i^s, t_i^p, t_i^o \rangle$ is a subject-predicate-object triple of RDF resources. The subject t_i^s of a triple t_i is an entity (i.e., a non-literal resource at the instance level) that appears in T. The predicate t_i^p represents a property. The object t_i^o is a value of t_i^p, which can be a class, a literal, or another entity in T.

A keyword query is a set of keywords denoted by $Q = \{q_1, q_2, \cdots, q_m\}$. Given a dataset T, a keyword query Q, and a positive integer k as the size bound, a *dataset snippet* is an optimum subset of triples selected from T, denoted by $S \subseteq T$, satisfying $|S| \leq k$. We will give our definition of optimality in Section 3.

2.2 Evaluation Metrics

We briefly review the four metrics proposed in [7] for evaluating the quality of a dataset snippet S: `coKyw`, `coCnx`, `coSkm`, and `coDat`, all in the range of $[0, 1]$.

Coverage of Query Keywords (`coKyw`). A resource r covers a keyword q if r's textual form (e.g., `rdfs:label` of an IRI or blank node, lexical form of a literal) contains a keyword match for q. A triple t covers a keyword q, denoted by $t \prec q$, if r covers q for any $r \in \{t^{\mathrm{s}}, t^{\mathrm{p}}, t^{\mathrm{o}}\}$. For a snippet S, the `coKyw` metric evaluates its coverage of query keywords:

$$\texttt{coKyw}(S) = \frac{1}{|Q|} \cdot |\{q \in Q : \exists t \in S,\ t \prec q\}| \,. \tag{1}$$

Coverage of Connections between Query Keywords (`coCnx`). A snippet S covers the connection between two keywords $q_i, q_j \in Q$, denoted by $S \prec (q_i, q_j)$, if there is a path in the RDF graph representation of S that connects two resources: one covering q_i and the other covering q_j. For S, the `coCnx` metric evaluates its coverage of connections between query keywords:

$$\texttt{coCnx}(S) = \begin{cases} \frac{1}{\binom{|Q|}{2}} \cdot |\{\{q_i, q_j\} \subseteq Q : q_i \neq q_j \text{ and } S \prec (q_i, q_j)\}| & \text{if } |Q| > 1\,, \\ \texttt{coKyw}(S) & \text{if } |Q| = 1\,. \end{cases} \tag{2}$$

When there is only one keyword, coCnx is meaningless and we set it to coKyw.

Coverage of Data Schema (coSkm). Consider the RDF schema of a dataset. The relative frequency of a class c observed in a dataset T is

$$\texttt{frqCls}(c) = \frac{|\{t \in T : \ t^{\text{p}} = \texttt{rdf:type} \text{ and } t^{\text{o}} = c\}|}{|\{t \in T : \ t^{\text{p}} = \texttt{rdf:type}\}|}. \tag{3}$$

Analogously, the relative frequency of a property p observed in T is

$$\texttt{frqPrp}(p) = \frac{|\{t \in T : \ t^{\text{p}} = p\}|}{|T|}. \tag{4}$$

For a snippet S, its coverage of the schema of T is the harmonic mean (hm) of the total relative frequency of the classes and properties it contains:

$$\texttt{coSkm}(S) = \texttt{hm}(\sum_{c \in \texttt{Cls}(S)} \texttt{frqCls}(c), \sum_{p \in \texttt{Prp}(S)} \texttt{frqPrp}(p)), \tag{5}$$

where $\texttt{Cls}(S)$ is the set of classes instantiated in S and $\texttt{Prp}(S)$ is the set of properties instantiated in S.

Coverage of Data (coDat). Central entities represent the key content of a dataset. Let $\texttt{d}^+(e)$ and $\texttt{d}^-(e)$ be the out-degree and in-degree of an entity e in the RDF graph representation of a dataset T, respectively. For a snippet S, its coverage of the entities in T is the harmonic mean (hm) of the mean normalized out-degree and in-degree of the entities it contains:

$$\begin{aligned}\texttt{coDat}(S) = \texttt{hm}(&\frac{1}{|\texttt{Ent}(S)|} \cdot \sum_{e \in \texttt{Ent}(S)} \frac{\log(\texttt{d}^+(e)+1)}{\max_{e' \in \texttt{Ent}(T)} \log(\texttt{d}^+(e')+1)}, \\ &\frac{1}{|\texttt{Ent}(S)|} \cdot \sum_{e \in \texttt{Ent}(S)} \frac{\log(\texttt{d}^-(e)+1)}{\max_{e' \in \texttt{Ent}(T)} \log(\texttt{d}^-(e')+1)}),\end{aligned} \tag{6}$$

where $\texttt{Ent}(X)$ is the set of entities that appear in a set of triples X.

3 Approach

Given the evaluation metrics presented in Section 2.2, a straightforward idea is to formulate the selection of RDF triples as a combinatorial optimization problem, and directly optimize these evaluation metrics. Our current work considers three metrics: coKyw, coSkm, and coDat, leaving coCnx as future work. The three selected metrics all require a snippet to cover some elements: query keywords in coKyw, classes and properties in coSkm, and entities in coDat. Furthermore, the classes, properties, and entities to cover are with different weights. It inspires us to formulate an instance of the weighted maximum coverage problem. We formalize this idea in Section 3.1, and present a solution in Section 3.2.

Algorithm 1 Greedy Algorithm

Input: A dataset T, a keyword query Q, and a size bound k
Output: An optimum dataset snippet $S \subseteq T$

```
S ← ∅;
while |S| < k do
    t* ← argmax_{t∈(T\S)} (q(S ∪ {t}) − q(S));
    S ← S ∪ {t*};
end while
return S;
```

3.1 Snippet Generation as Weighted Maximum Coverage

Weighted Maximum Coverage. Given a collection of sets, a weighted maximum coverage (WMC) problem is to select a limited number of sets from the collection such that the total weight of the covered elements is maximized.

Snippet Generation as WMC. We formulate the generation of an optimum dataset snippet as an instance of the WMC problem. Each RDF triple $t_i \in T$ corresponds to a set denoted by $\texttt{cov}(t_i)$ which consists of: the query keywords covered by t_i, the class instantiated in t_i, the property instantiated in t_i, and the entities that appear in t_i. The universe of elements is denoted by

$$\Omega = Q \cup \texttt{Cls}(T) \cup \texttt{Prp}(T) \cup \texttt{Ent}(T)\,. \tag{7}$$

Each element $x \in \Omega$ has a non-negative weight:

$$\texttt{w}(x) = \begin{cases} \alpha \cdot \frac{1}{|Q|} & x \in Q\,, \\ \beta \cdot \texttt{frqCls}(x) & x \in \texttt{Cls}(T)\,, \\ \beta \cdot \texttt{frqPrp}(x) & x \in \texttt{Prp}(T)\,, \\ \gamma \cdot \left(\frac{\log(\texttt{d}^+(x)+1)}{\sum_{e \in \texttt{Ent}(T)} \log(\texttt{d}^+(e)+1)} + \frac{\log(\texttt{d}^-(x)+1)}{\sum_{e \in \texttt{Ent}(T)} \log(\texttt{d}^-(e)+1)}\right) & x \in \texttt{Ent}(T)\,, \end{cases} \tag{8}$$

In our experiments, we set $\alpha = 2, \beta = 1, \gamma = 1$, to balance between the coverage of query keywords in `coKyw` (α), the coverage of classes and properties in `coSkm` (β), and the coverage of entities in `coDat` (γ) in our objective function.

An optimum dataset snippet $S \subseteq T$ is one that

$$\text{maximizes } \texttt{q}(S) = \sum_{x \in \bigcup_{t_i \in S} \texttt{cov}(t_i)} \texttt{w}(x)\,, \quad \text{subject to } |S| \leq k\,, \tag{9}$$

where k is a predefined size bound, and $\texttt{q}(\cdot)$ is the objective function.

3.2 Solution

Algorithm 1 presents the greedy algorithm for the WMC problem which at each stage chooses a set that contains the maximum weight of uncovered elements. It achieves an approximation ratio of $1 - \frac{1}{e}$.

	coKyw	coCnx	coSkm	coDat	Average
IlluSnip	0.1000	0.0540	0.6820	0.3850	0.3053
TA+C	0.9590	0.4703	0.0425	0.0915	0.3908
PrunedDP++	1	1	0.0898	0.2133	0.5758
CES	0.9006	0.3926	0.3668	0.2684	0.4821
KSD	0.8352	0.3595	0.8651	0.4247	0.6211

Table 1: Average scores of different methods over all the query-dataset pairs.

	coKyw	coCnx	coSkm	coDat	Average
data.gov.uk	0.7643	0.2882	0.8249	0.3870	0.5661
DMOZ-1	0.8977	0.7955	0.8873	0.4726	0.7633
DMOZ-2	0.8433	0.2444	0.8710	0.4569	0.6039
DMOZ-3	0.8395	0.2337	0.8693	0.4145	0.5893
DMOZ-4	0.7936	0.1877	0.8521	0.3731	0.5516

Table 2: Average scores of KSD over each group of query-dataset pairs.

Assuming $\mathtt{q}(S \cup \{t\}) - \mathtt{q}(S)$ is computed in $O(1)$, the overall running time of a naive implementation of the algorithm is $O(k \cdot n)$, where n is the number of RDF triples in T. A more efficient implementation may use a priority queue to hold candidate triples, which is left as our future work.

4 Evaluation

Our evaluation reused the 387 query-dataset pairs in [7] where datasets were collected from DataHub and queries included 42 real queries submitted to data.gov.uk and 345 artificial queries comprising i category names in DMOZ referred to as DMOZ-i for $i = 1, 2, 3, 4$. We compared our proposed KSD with four baseline methods evaluated in [7], namely IlluSnip [2], TA+C [5], PrunedDP++ [6], and CES [4]. Following [7], we set $k = 20$, i.e., a snippet contained at most 20 triples.

4.1 Quality of Snippets

Table 1 presents the average scores of the four evaluation metrics over all the query-dataset pairs. Compared with the baselines, KSD achieved the highest overall score of 0.6211. In particular, its coverage of schema ($\mathtt{coSkm} = 0.8651$) and data ($\mathtt{coDat} = 0.4247$) were at the top. Its coverage of query keywords ($\mathtt{coKyw} = 0.8352$) was close to TA+C, PrunedDP++, and CES which are query-focused methods. Therefore, KSD achieved a satisfying trade-off between these evaluation metrics. On the other hand, its coCnx score was not high because coCnx was not explicitly considered in our approach.

Table 2 breaks down the scores of KSD into groups of query-dataset pairs. The scores on different groups were generally consistent with each other, demonstrating the robustness of our approach. One exception was the very high coCnx score on DMOZ-1, due to Eq. (2) where $\mathtt{coCnx} = \mathtt{coKyw}$ when $|Q| = 1$.

4.2 Running Time

We tested the running time of our approach on an Intel Core i7-8700K (3.70GHz) with 10GB memory for the JVM.

Among all the 387 query-dataset pairs, for 234 (60.47%) a dataset snippet was generated within 1 second, and for 341 (88.11%) one was generated within 10 seconds. The median time was 0.51 second, showing promising performance

for practical use. In the worst case, it took 150 seconds to process a large dataset containing more than 2 million RDF triples. Future work would be needed to improve the performance of our implementation to handle large datasets.

5 Conclusion and Future Work

In this ongoing work, we proposed KSD, a new approach to generating snippets for dataset search. By directly optimizing three evaluation metrics, KSD outperformed four baselines. It has established new state-of-the-art results for future work. We are working towards a full version of KSD which will also optimize `coCnx`. We will implement our approach in a prototype of a new dataset search engine, to help users conveniently judge the relevance of a retrieved dataset.

There are limitations in our work. First, the current version of KSD has considered three metrics but we exclude `coCnx`. The other three metrics are all about covering some elements with selected RDF triples, whereas `coCnx` is related to graph connectivity. The weighted maximum coverage problem seems not expressive enough to model `coCnx`. We will explore other possibilities. Second, although the running time of our current implementation is acceptable in most cases, its performance is not satisfying on large datasets. We will consider using priority queue and appropriate indexes to make the generation process faster.

Acknowledgements

This work was supported by the NSFC under Grant 61572247. Cheng was funded by the Six Talent Peaks Program of Jiangsu Province under Grant RJFW-011.

References

1. Brickley, D., Burgess, M., Noy, N.F.: Google dataset search: Building a search engine for datasets in an open web ecosystem. In: WWW 2019. pp. 1365–1375 (2019)
2. Cheng, G., Jin, C., Ding, W., Xu, D., Qu, Y.: Generating illustrative snippets for open data on the web. In: WSDM 2017. pp. 151–159 (2017)
3. Ellefi, M.B., Bellahsene, Z., Breslin, J.G., Demidova, E., Dietze, S., Szymanski, J., Todorov, K.: RDF dataset profiling - a survey of features, methods, vocabularies and applications. Semantic Web **9**(5), 677–705 (2018)
4. Feigenblat, G., Roitman, H., Boni, O., Konopnicki, D.: Unsupervised query-focused multi-document summarization using the cross entropy method. In: SIGIR 2017. pp. 961–964 (2017)
5. Ge, W., Cheng, G., Li, H., Qu, Y.: Incorporating compactness to generate term-association view snippets for ontology search. Inf. Process. Manage. **49**(2), 513–528 (2013)
6. Li, R., Qin, L., Yu, J.X., Mao, R.: Efficient and progressive group steiner tree search. In: SIGMOD 2016. pp. 91–106 (2016)
7. Wang, X., Chen, J., Li, S., Cheng, G., Pan, J., Kharlamov, E., Qu, Y.: A framework for evaluating snippet generation for dataset search. In: ISWC 2019 (2019), https://arxiv.org/abs/1907.01183