Populating Personal Linked Data Caches using Context Models

Olaf Hartig
Humboldt-Universität zu Berlin
Unter den Linden 6, 10099 Berlin, Germany
hartig@informatik.hu-berlin.de

Tom Heath
Talis Education Ltd.
43 Temple Row, Birmingham, B2 5LS, UK
tom.heath@talis.com

ABSTRACT

The emergence of a Web of Data [3] enables new forms of application that require expressive query access, for which mature, Web-scale information retrieval techniques may not be suited. Rather than attempting to deliver expressive query capabilities at Web-scale, this paper proposes the use of smaller, pre-populated data caches whose contents are personalized to the needs of an individual user. We present an approach to *a priori* population of such caches with Linked Data harvested from the Web, seeded by a simple context model for each user, which is progressively enriched by executing a series of *enrichment rules* over Linked Data from the Web. Such caches can act as personal data stores supporting a range of different applications. A comprehensive user evaluation demonstrates that our approach can accurately predict the relevance of attributes added to the context model and the execution probability of queries based on these attributes, thereby optimizing the cache population process.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous; K.8.m [**Personal Computing**]: Miscellaneous

Keywords

Context, Query Prediction, Cache Population, Linked Data

1. INTRODUCTION

The emergence of a Web of Data [3] enables new forms of application that require expressive query access, for which mature, Web-scale information retrieval techniques may not be suited. It is also not apparent that providing expressive query access, e.g. SPARQL, over caches of *all Linked Data* available on the Web is feasible long-term, due to the computational cost and the absence of meaningful ranking functions.

Therefore, we propose an alternative strategy: *a priori* construction of personalized data sets, i.e. building many small caches of data, each personalized to a particular user. These *personal caches* can prevent application-specific data silos (by acting as a data backend to multiple applications) and enable new forms of search functionality (e.g. complex, long-running and non-time-critical queries) by allowing the cache owner to control resource allocation.

Our approach to personalization is based on the observation that a user's information needs are a function of their context, but information needs themselves will often map to a finite set of templated queries. Therefore, given a description of an upcoming context, we aim to pro-actively populate a cache with data relevant to that context. In order to ensure the relevance of the cached data it is necessary to predict those queries that are most likely in the given context. In this paper we introduce an approach to context-dependent *query prediction*, the results of which can be used to automatically populate a personal cache with relevant data.

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

2. OUR APPROACH

In summary, the goal of our approach is a cache populated with data relevant to queries which may be issued by an application when used in a particular context. To achieve this goal we predict those queries that are most likely to be issued in that context – these queries can then be used to discover and to select data for the cache. This query-based data discovery may be realized by executing the queries over the Web (e.g. using a link traversal based execution system [2, 1]) or by accessing existing Linked Data indexes.

One potential obstacle to our approach is sparse descriptions of a particular context. To address this issue we enrich the context description with additional, implicit attributes. Consequently, our approach includes three steps – context enrichment, query prediction, and cache population – depicted in Figure 1.

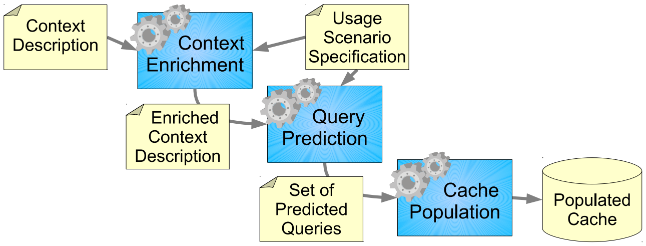

Figure 1: The general workflow of our context-based cache population approach.

2.1 Context Enrichment

The context enrichment process begins with two inputs: a *context description* and a *usage scenario specification*. The usage scenario specification consists of two artifacts: 1) rules for enriching relevant aspects of a context description, and 2) a description of relevant types of user tasks and of the way an application supports these tasks with the execution of prepared queries.

The context description describes a context in the form of a set of context attributes, such as a user's current location or their strong interest in gothic architecture, and a relevancy score which denotes the degree to which the attributes are relevant in that context.

Each context attribute in the context description is described in the form of RDF triples. In order to increase the coverage of a context by a context description, we use a *context enrichment* process that infers additional CAs based on SPARQL graph patterns. For example, a context description that includes the context attribute *Gothic Architecture* may be enriched with context attributes based on broader topics such as *Gothic Art*

2.2 Context-Aware Task Model

Our context-aware task model formally captures those user tasks which are supported by software applications, and characterizes these in the form of *context dependent properties* (CDP) and query templates, all described in RDF. These descriptions present the basis for a *usage scenario specification* and which collectively serve as input to the query prediction step.

2.3 Query Prediction

The second step in our cache population process (cf. Figure 1) comprises the prediction of SPARQL queries that applications will execute over the personal cache in a given, future context. Each of these queries is associated with an *execution probability* which represents the likelihood that the query is actually executed in the given context.

The general idea of our approach is to generate all possible instantiations of those query templates that applications may instantiate in a future context given by an (enriched) context description, and then estimate an execution probability for each generated query.

2.4 Cache Population

The final step of our cache population process consists of executing a query based population method, the goal of which is the creation of a cache of Linked Data that enables the query system to answer the given set of queries as effectively as possible. To achieve this goal the population method requires two strategies: i) a strategy for discovering potentially relevant Linked Data on the Web and ii) a strategy for selecting the most suitable subset of discovered data for the cache.

While a precise definition and an analysis of such strategies is not the main focus of this paper, approaches based on crawling, use of existing Linked Data indexes, or *query execution-based approaches* (such as [2, 1]) may all be viable.

In terms of data selection strategy, the goal is to find a subset of the data that fits into the cache and that guarantees maximum utility. An approach to achieve this goal may be based on the (estimated) execution probabilities of predicted queries: data that has been discovered based on queries with a high execution probability has the highest priority for the cache.

3. EVALUATION

In the final workflow step (see Figure 1), the set of *predicted queries* is used as input to the *cache population* process. Therefore, the ability to populate the cache with the most relevant data is dependent on the effectiveness of the query prediction strategy, specifically the ability to accurately predict the execution probability of a query.

We evaluated the accuracy of our prediction of execution probabilities by assessing their degree of correlation with participants' ratings of the likelihood of executing each query; we refer to the *per query* aggregate of these ratings as *actual execution probabilities*. This was achieved by presenting participants with a concrete description of a *stranded traveller* scenario and asked to rate (on a scale of 0-10) the likelihood that, if stranded at a specific airport with access to our hypothetical application, they would ask various questions involving certain locations near the airport. The degree of correlation between the predicted and actual execution probabilities was calculated on a per-airport basis (and for each permutation of weights and aggregation functions used in computing these probabilities) using *Spearman's rank correlation coefficient*, as summarized in Table 1, which shows the highest and lowest values of ρ for all queries across each airport. Comparison of each ρ to the critical values (taken from [4]) in the final column shows that all permutations of weights and aggregation functions produced correlations that are statistically significant at the 5% *alpha level* (α=*0.05*), for all airports. Therefore, we conclude that our approach is able to predict, with a high degree of accuracy, the actual execution probability of queries instantiated with a wide range of values in the stranded traveller scenario.

Airport	***N***	**Lowest ρ**	**Highest ρ**	**Crit. Val. at α=0.05**
Coleman	151	0.216	0.328	0.165
Edmonton	122	0.289	0.566	0.165
Halifax	86	0.321	0.660	0.179

Table 1: Highest and lowest values of ρ for all queries across each group.

4. CONCLUSIONS AND FUTURE WORK

In this paper we introduced the concept of personal caches, populated with Linked Data from the Web based on the user's context. We introduced a model for context representation and enrichment that complements our context-aware task model, in order to predict queries that an application may issue in response to user information needs in a particular context. Central to this approach is accurately predicting the probability of a particular query being executed, as this influences the relevance of data in the cache. We demonstrate our ability to predict these execution probabilities across a range of contexts, thereby supporting our context-based approach as a basis for populating a cache with relevant data.

Future work on this topic has two main areas of focus: the feasibility of forecasting a future context and it's attributes, based, for example, on a user's personal schedule; secondly, further investigation is required to select the most appropriate cache population method.

5. REFERENCES

[1] O. Hartig. How caching improves efficiency and result completeness for querying linked data. In *Proc. of the 4th Int. Workshop on Linked Data on the Web*, 2011.

[2] O. Hartig, C. Bizer, and J.-C. Freytag. Executing SPARQL queries over the Web of Linked Data. In *Proc. of the 8th Int. Semantic Web Conf.*, 2009.

[3] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.

[4] P. H. Ramsey. Critical values for spearman's rank order correlation. *Journal of Educational Statistics*, 14(3), 1989.