Processamento de consultas na Web de Dados: uma abordagem para busca de fontes de dados relevantes

Alberto Trindade Tavares, Bernadette Farias Lóscio

Centro de Informática - Universidade Federal de Pernambuco, Brasil
{att, bfl}@cin.ufpe.br

Abstract. The adoption of Linked Data principles has contributed towards the creation of a Web of Data, allowing the development of applications and tools which run queries over available information. One of the main challenges for the query processing over the Web is the selection of relevant sources, i.e., sources which could contribute significantly to the result of a query. In this paper, we discuss this problem and present an approach for identifying Web sources that may potentially be relevant to the processing of a set of queries. A distinct issue of our work is that the process of searching for sources employs the user requirements expressed in SPARQL queries.

Resumo. A adoção dos princípios do *Linked Data* tem contribuído para a construção de uma Web de Dados, permitindo o desenvolvimento de aplicativos e ferramentas que executam consultas sobre as informações disponibilizadas. Diante do crescente volume de dados desta natureza, um dos principais desafios para o processamento de consultas sobre a Web é a seleção de fontes relevantes, ou seja, aquelas capazes de contribuir de maneira significativa com os resultados de uma determinada consulta. Neste artigo, discutimos este problema e propomos uma abordagem para identificar fontes da Web de Dados que possam, potencialmente, contribuir com os resultados de um conjunto de consultas. Uma característica importante da abordagem apresentada é que o processo de busca de fontes faz uso dos requisitos de usuário expressos em consultas SPARQL.

Categories and Subject Descriptors: H.4 [**Information System Applications**]: Miscellaneous; H.2 [**Database Management**]: Miscellaneous

Keywords: Semantic Web, Linked Data, Web Crawling.

1. INTRODUÇÃO

A Web, por meio da publicação de documentos de hipertexto, estabeleceu um espaço global de informações. No entanto, a atual estrutura da mesma apresenta diversas limitações quanto ao processamento automático do seu conteúdo, pois os documentos são especificados em linguagens de marcação que fornecem, essencialmente, descrições sintáticas. Para lidar com esta restrição, diversas iniciativas foram desenvolvidas para facilitar o compartilhamento e processamento dos dados publicados. Dentre essas iniciativas, destaca-se a Web Semântica (*Semantic Web*), uma extensão da Web atual, onde os dados estão associados a um significado compreensível por máquinas. No contexto da Web Semântica, o termo *Linked Data* é utilizado para descrever um conjunto de práticas para a publicação de dados estruturados na Web, utilizando o modelo RDF para a representação do conhecimento.

A adoção dos padrões de *Linked Data* tem levado à construção de uma Web de Dados, caracterizado como um grafo global de dados formado por bilhões de triplas RDF. As informações disponibilizadas neste ambiente cobrem uma vasta gama de tópicos, tais como localizações geográficas, pessoas, empresas, livros, publicações científicas, dados estatísticos, entre outros [Bizer et al. 2009]. A publicação de dados interligados tem motivado o desenvolvimento de aplicações e ferramentas, pois a viabilidade de consultar

esta nuvem de dados, como se eles constituíssem um grande banco de dados distribuído, oferece diversas possibilidades. Entretanto, executar consultas sobre a Web de Dados ainda é um desafio para os desenvolvedores [Hartig 2013].

Neste trabalho, estamos interessados nas aplicações que acessam fontes de dados RDF, publicados de acordo com os princípios de *Linked Data*, onde este acesso é feito por meio de consultas escritas na linguagem SPARQL. Especificamente, este artigo aborda o problema da identificação de fontes, disponíveis na Web de Dados, relevantes para o processamento de um conjunto de consultas SPARQL. São consideradas fontes de dados relevantes, aquelas que podem contribuir com informações úteis do ponto de vista do usuário [Oliveira et al. 2012], ou seja, atendem aos requisitos dos usuários. Na abordagem proposta, consideramos que os requisitos de usuário são expressos nos padrões de triplas das consultas. O uso desta abordagem permite a detecção de fontes de dados inicialmente não conhecidas, mas que, potencialmente, irão contribuir com os resultados de consultas de uma aplicação.

O restante deste artigo é organizado como se segue. Na Seção 2, são apresentados os principais conceitos e termos relacionados às tecnologias da Web Semântica, fornecendo a fundamentação teórica para este trabalho. A Seção 3 descreve a abordagem proposta para a busca de fontes RDF na Web. A seção 4 mostra um exemplo prático de utilização da abordagem. Por fim, a Seção 5 conclui o artigo, citando contribuições deste trabalho e indicando pesquisas futuras.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, são apresentados alguns conceitos e terminologias que serão utilizados ao longo deste artigo.

2.1 URI e RDF

URI[1] é uma sequência de caracteres que identifica unicamente um recurso Web. O mecanismo básico para acessar recursos Web, segundo os padrões de *Linked Data*, se dá através de um processo chamado de dereferenciamento de URIs, que consiste no acesso via HTTP a uma URI, obtendo-se um conjunto de descrições RDF [Bizer et al. 2009]. O RDF[2], por sua vez, é um modelo de dados que permite descrever recursos na Web por meio de triplas, as quais podem ser organizadas como grafos direcionados. Os três componentes de uma tripla são: *Sujeito*, *Predicado* e *Objeto*.

2.2 SPARQL

SPARQL[3] é uma linguagem de consulta para dados RDF, permitindo a recuperação de informação contida em grafos. As principais partes de uma consulta SPARQL são [Pérez et al. 2009]: o *padrão da consulta*, que é composto por um conjunto de padrões de triplas (*Triple Patterns*), constituindo o denominado BGP (*Basic Graph Pattern*) da consulta, responsável por descrever os padrões com os quais as triplas resultantes devem estabelecer correspondência; os *modificadores de solução*, que permitem reorganizar o resultado da consulta e a *saída*, que especifica o formato do resultado. Fontes de dados interligados tipicamente disponibilizam um SPARQL *endpoint*[4], um serviço Web que permite ao usuário submeter consultas SPARQL sobre os dados RDF armazenados na fonte.

[1]http://www.w3.org/TR/uri-clarification/
[2]http://www.w3.org/RDF/
[3]http://www.w3.org/TR/rdf-sparql-query/
[4]http://semanticweb.org/wiki/SPARQL_endpoint

2.3 Web Crawler

A extração de dados na Web pode ser realizada através de Web *crawlers*, agentes de software que acessam a Web de maneira automatizada, navegando entre os recursos por meio de *links* [Castillo 2005]. A tarefa realizada por este agente é chamada de *crawling*. Um *crawler* inicia sua busca de dados na Web a partir de um conjunto de recursos de origem, denominados s*eeds* [Tavares et al. 2012a].

3. A ABORDAGEM PROPOSTA

Diante do número crescente de fontes de dados interligados que estão disponíveis na Web, se torna um desafio a execução de consultas sobre a Web de Dados. Uma das dificuldades nesta tarefa é a seleção das fontes que são capazes de responder a essas consultas. Neste trabalho, propomos uma abordagem que realiza um filtro nos conjuntos de dados, disponíveis na Web de Dados, usando como critério de seleção os requisitos de usuários que podem ser extraídos de consultas SPARQL. Especificamente, os requisitos de usuário considerados nesta abordagem são aqueles que podem ser extraídos a partir do BGP de uma consulta SPARQL e correspondem às URIs que representam um recurso (sujeito ou objeto) ou um predicado presente nos padrões de triplas do BGP.

A abordagem de busca de fontes proposta pode ser dividida em três etapas: i) Extração de recursos mais relevantes; ii) Realização de um *crawling* na Web para detectar fontes de dados interligados que descrevam os principais recursos extraídos na etapa anterior e iii) Classificação das fontes detectadas de acordo com a taxa de cobertura dos predicados das consultas da aplicação.

O Algoritmo *DatasetsSearch* (Algoritmo 1) apresenta o processo de busca proposto, descrevendo as etapas listadas acima. O algoritmo recebe, como entrada, um conjunto *Q* de consultas SPARQL e fornece, como saída, uma lista de fontes de dados, seguindo uma ordem de relevância, que são potencialmente capazes de responder as consultas em *Q*. Especificamente, a saída é composta por uma lista de SPARQL *endpoints*. O Algoritmo *DatasetsSearch* é detalhado nas subseções a seguir.

Algorithm DatasetsSearch
Input Q: A set of SPARQL queries
  k: Number of relevant resources to be considered during the crawling
Output *DE:* A list of SPARQL endpoints of fetched datasets
Begin
1. $RR \leftarrow \boldsymbol{ExtractRelevantResources}(Q)$
2. $Seeds \leftarrow SelectResources(RR, k)$
3. $Predicates \leftarrow \{sameAs,\ seeAlso,\ equivalentClass\}$
4. $FetchedTriples \leftarrow ExecuteCrawling(Seeds, Predicates)$
5. $ProvenanceTriples \leftarrow ExtractProvenance(FetchedTriples)$
6. $Endpoints \leftarrow \emptyset$
7. $\boldsymbol{For\ each}\ p \in ProvenanceTriples\ \boldsymbol{do}$
8. $\quad Endpoints \leftarrow Endpoints \cup RetrieveSparqlEndpoint(p)$
9. $\boldsymbol{End\ for}$
10. $QueryPredicates \leftarrow ExtractPredicates(Q)$
11. $DE \leftarrow \boldsymbol{RankDatasetsByPredicates}(Endpoints, QueryPredicates)$
12. $\boldsymbol{Return}\ DE$

End

Algoritmo 1. Algoritmo para Busca de Fontes de Dados

3.1 Extração de Recursos Relevantes

O primeiro passo da busca de fontes de dados é a identificação dos recursos mais relevantes a partir das consultas fornecidas como entrada. Consideramos, neste trabalho, que os recursos mais relevantes são os

mais frequentes no BGP da consulta. Estes recursos irão guiar a busca por fonte de dados na Web. A identificação é realizada através da função *ExtractRelevantResources*, apresentada no Algoritmo 2, que recebe como entrada um conjunto de consultas *Q* e retorna a lista dos recursos mais frequentes de *Q*. Especificamente, a função *ExtractRelevantResources* recupera o BGP de cada uma das consultas em *Q* e, para cada padrão de tripla de um dado BGP, seus elementos (sujeito, predicado e objeto) são processados por um *Visitor*. Ao longo da navegação desses elementos, temos a construção de uma lista de recursos, que estão presentes nos padrões de triplas das consultas, e de suas respectivas quantidades de ocorrência. A partir dessa lista, é gerada uma segunda lista, *RR* (*RelevantResources*), que armazena os recursos ordenados pela frequência, de forma decrescente.

3.2 Web Crawling para a Busca de Fontes

Uma vez que os recursos mais frequentes são identificados, o próximo passo é a execução de um *crawling* na Web de Dados, com o objetivo de buscar fontes relevantes para a execução das consultas. O processo de *crawling* considera como *seeds* os *k* primeiros recursos da lista *RR*, constituída por URIs que representam os recursos relevantes do conjunto de consultas *Q*. O conjunto de predicados {*rdfs:seeAlso*, *owl:sameAs* e *owl:equivalentClass*} é definido como outro parâmetro do *crawling*, indicando os *links* a serem seguidos pelo agente. O uso desses predicados permite a obtenção de novos recursos da Web que são similares aos recursos *seeds* [Ding et al. 2010].

Algorithm ExtractRelevantResources

Input Q: A set of queries
Output *RR:* A sorted list by frequency of query resources
Begin
1. $FrequencyList \leftarrow \emptyset$
2. ***For each*** $q \in Q$ ***do***
3. $BGP \leftarrow ExtractBGP(q)$
4. ***For each*** $triplePattern \in BGP$ ***do***
5. $Resources \leftarrow VisitTriplePattern(triplePattern)$
6. $FrequencyList \leftarrow FrequencyList \cup Resources$
7. ***End for***
8. ***End for***
9. $RR \leftarrow DecreasingElementsList(FrequencyList)$
10. ***Return*** RR

End

Algoritmo 2. Algoritmo para Extração de Recursos Relevantes

Algorithm RankDatasetsByPredicates

Input SE: A set of SPARQL endpoints
QP: A list of predicates
Output DE: A sorted list of SPARQL endpoints
Begin
1. $PredicatesRateByEndpoint \leftarrow NewMap()$
2. ***For each*** $e \in SE$ ***do***
3. $PR \leftarrow RetrievePredicatesRate(e, QP)$
4. $PutKeyValueInMap(e, PR, PredicatesRateByEndpoint)$
5. ***End for***
6. $DE \leftarrow GetKeysSortedByValue(PredicatesRateByEndpoint)$
7. ***Return*** RR

End

Algoritmo 3. Algoritmo para Classificação das Fontes Recuperadas

Ao final do *crawling*, temos um conjunto de triplas extraídas da Web de Dados que descrevem os recursos selecionados como *seeds*. O próximo passo do algoritmo é a construção da lista das fontes descobertas ao longo da busca. Para esta tarefa, são extraídas as proveniências das triplas coletadas pelo Web *crawler*. A informação de proveniência de uma tripla é representada por uma URI que indica a localização do seu arquivo RDF de origem. Para cada URI de proveniência, é utilizada a função *RetrieveSparqlEndpoint*, responsável por recuperar a URI do SPARQL *endpoint* das fontes de dados de procedência das triplas. O número de fontes, identificadas nesta etapa, é determinado pela quantidade de *links rdfs:seeAlso*, *owl:sameAs* e *owl:equivalentClass* subsequentes existentes a partir das fontes de origem do *crawling*, as quais são procedentes do dereferenciamento das URIs *seeds*.

3.3 Classificação das Fontes Detectadas

A última etapa do algoritmo é a classificação das fontes detectadas na busca de acordo com a taxa de cobertura dos predicados das consultas. Para uma determinada fonte de dados, esta taxa mede a porcentagem dos predicados presentes nos padrões de triplas das consultas *Q* que aparecem em triplas RDF da respectiva fonte. Essa métrica é utilizada por permitir a avaliação das fontes quanto à capacidade de satisfazer os padrões de correspondência definidos nas consultas. A classificação é realizada pela função *RankDatasetsByPredicates*, apresentada no Algoritmo 3. A função recebe, como parâmetros, o conjunto de SPARQL *endpoints* obtidos na etapa anterior e os predicados das consultas *Q*, que são extraídos por meio da função *ExtractPredicates*.

A função *RankDatasetsByPredicates* aplica, para cada um dos *endpoints*, a função *RetrievePredicatesRate* que irá calcular a taxa de cobertura de predicados de *Q* para a fonte associada ao *endpoint*. Esta função executa uma consulta SPARQL, através do *endpoint* dado como parâmetro, para cada um dos predicados das consultas, verificando se existe alguma tripla que possua o respectivo predicado. Por fim, os *endpoints* são ordenados segundo a taxa de cobertura de predicados, de forma decrescente, gerando a lista *DE* (*Dataset Endpoints*) que é fornecida como saída do Algoritmo 3. A lista de SPARQL *endpoints* retornada pela função *RankDatasetsByPredicates* é o resultado final do algoritmo *DatasetsSearch*. Os *endpoints* provêm à aplicação o acesso a fontes da Web de Dados, permitindo a execução das consultas SPARQL sobre as mesmas.

4. EXEMPLO DE UTILIZAÇÃO

Para ilustrar a abordagem proposta neste trabalho, esta seção apresenta um exemplo de utilização sobre o domínio bibliográfico. Para este domínio, diversas fontes estão disponíveis na Web de Dados fornecendo informações a respeito de autores, conferências, artigos, livros, entre outros tópicos relacionados. Considere três consultas SPARQL sobre este domínio, selecionando informações sobre o autor *Alon Halevy* e o evento *International Semantic Web Conference* (ISWC). Essas consultas são mostradas nas figuras 1, 2 e 3.

Q1. Retorne o título de artigos que foram publicados na ISWC 2012	*Q2. Retorne o nome dos autores que tiveram artigos publicados na ISWC 2012*	*Q3. Retorne o título de artigos que foram escritos por Alon Y. Halevy*
SELECT ?tituloArtigo WHERE *{* *{ ?artigo akt:cites-publication-reference id:conf/iswc/2012 . }* *{ ?artigo akt:has-title ?tituloArtigo . }* *}*	*SELECT DISTINCT ?nomeAutor WHERE* *{* *{ ?artigo akt:cites-publication-reference id:conf/iswc/2012 . }* *{ ? artigo akt:has-title ?tituloArtigo . }* *{ ?artigo akt:has-author ?autor . }* *{ ?autor akt:full-name ? nomeAutor . }}*	*SELECT ?tituloArtigo WHERE* *{* *{ ?artigo akt:has-author id:people-a86143 . }* *{ ? artigo akt:has-title ?tituloArtigo . }* *}*
Figura 1. Consulta #1	**Figura 2. Consulta #2**	**Figura 3. Consulta #3**

Para buscar fontes na Web de Dados que possam responder a essas consultas, vamos aplicar as três etapas da abordagem. A primeira tem o objetivo de selecionar os recursos relevantes dessas consultas, ou seja, URIs que representam um sujeito ou objeto e que fazem parte de algum dos três padrões de consultas considerados. A Tabela 1 apresenta o resultado desta fase, onde dois recursos são extraídos dos padrões de triplas das consultas e classificados de acordo com a frequência. A etapa seguinte consiste no *crawling* na

Web a partir dos dois recursos selecionados. Como resultado deste processo, são retornadas quatro fontes de dados, apresentadas na Tabela 2, juntamente com a URI do respectivo SPARQL *endpoint*. Entre essas fontes, a *DBLP RKBExplorer* é a única fonte procedente dos recursos *seeds*, sendo obtida no primeiro passo do processo de *crawling*. Por outro lado, as outras três fontes são descobertas ao longo da navegação entre os *links* RDF.

Tabela 1. Recursos Relevantes

URI do Recurso	# de Ocorrências
id:conf/iswc/2012	2
id:people-a86143	1

Tabela 2. Fontes Retornadas pelo Web *Crawler*

Nome da Fonte	URI do SPARQL *Endpoint*
DBLP RKBExplorer	http://dblp.rkbexplorer.com/sparql/
DBLP L3S	http://dblp.l3s.de/d2r/sparql
IEEE	http://ieee.rkbexplorer.com/sparql/
BibSonomy	-

Para a última etapa da abordagem, são consideradas somente as fontes de dados que disponibilizam um SPARQL *endpoint*, pois a interface para a execução de consultas sobre a base é fornecida por este serviço. Consequentemente, o *BibSonomy* é descartado, por não oferecer um *endpoint*, enquanto as outras três fontes são classificadas. A classificação das fontes é dada de acordo com a taxa de cobertura dos seguintes predicados: *akt:cites-publication-reference*, *akt:has-title*, *akt:has-author* e *akt:full-name*. Como resultado, as três fontes são igualmente classificadas, pois as mesmas possuem, entre as suas triplas RDF, todos os quatro predicados. O SPARQL *endpoint* dessas bases de dados são retornadas como a saída do Algoritmo, permitindo, a uma aplicação, a execução das consultas sobre tais fontes.

5. CONCLUSÃO

Neste artigo, apresentamos uma abordagem para a busca de fontes da Web de Dados que são, potencialmente, capazes de contribuir com os resultados de consultas de uma determinada aplicação. Essa abordagem faz uso dos requisitos de usuário expressos nas próprias consultas, que estabelecem o escopo da busca a ser executada. Como trabalhos futuros, gostaríamos de destacar algumas direções:

(i) Extensão de um protótipo desenvolvido para a busca de fontes [Tavares et al. 2012b], dando suporte ao uso de predicados de consultas para a classificação das fontes.

(ii) Realização de experimentos utilizando a nova versão do protótipo, para fins de validação da abordagem.

REFERÊNCIAS

BIZER, C., HEATH, T. and BERNERS-LEE, T. 2009. Linked data - the story so far. In *Proceedings of the International Journal on Semantic Web and Information Systems*, 5(3), 1-22.

HARTIG, O. 2013. An Overview on Execution Strategies for Linked Data Queries. In *Datenbankspektrum*, 13(2).

PÉREZ, J., ARENAS, M. and GUTIERREZ, C. 2009. Semantics and complexity of SPARQL. In *Proceedings of the ACM Transactions on Database Systems*, TODS 2009, Nova York, NY, USA, 34(3).

OLIVEIRA, H. R., TAVARES, A. T. and LÓSCIO, B. F. 2012. Feedback-based Data Set Recommendation for Building Linked Data Applications. In *Proceedings of the International Conference on Semantic Systems*, I-SEMANTICS 2012, Graz, Austria.

TAVARES, A. T., OLIVEIRA, H. R. and LÓSCIO, B. F. 2012. RDFMat – Um serviço para criação de repositórios de dados RDF a partir de crawling na Web de dados. In *I Escola Regional de Informática de Pernambuco*, 2012, Recife, Brasil.

CASTILLO, CARLOS. 2005. Effective Web Crawling. In ACM SIGIR Forum, Vol.39, Nova York, NY, USA.

DING L, SHINAVIER J, SHANGGUAN Z, MCGUINNESS DL. 2010. SameAs networks and beyond: analyzing deployment status and implications of owl: sameAs in linked data. In *Proc of the 9th International Semantic Web Conference*, ISWC 2010, Shanghai, China.

TAVARES, A. T., OLIVEIRA, H. R. and LÓSCIO, B. F. 2012. *Buscando Fontes de Dados Relevantes para Aplicações Linked Data. XVIII Simpósio Brasileiro de Sistemas Multimídia e Web*, WebMedia 2012, São Paulo. IX Workshop de Trabalhos de Iniciação Científica, 2012, p. 119-122.