Buscando Fontes de Dados Relevantes para Aplicações Linked Data

Alberto Trindade Tavares
Centro de Informática - Universidade Federal de Pernambuco
Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária, 50.740-560 - Recife – PE, Brasil
+55 81 2126.8430
att@cin.ufpe.br

Hélio Rodrigues de Oliveira
Centro de Informática - Universidade Federal de Pernambuco
Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária, 50.740-560 - Recife – PE, Brasil
+55 81 2126.8430
hro@cin.ufpe.br

Bernadette Farias Lóscio
Centro de Informática - Universidade Federal de Pernambuco
Av. Jornalista Anibal Fernandes, s/n
Cidade Universitária, 50.740-560 - Recife – PE, Brasil
+55 81 2126.8430
bfl@cin.ufpe.br

ABSTRACT
The growing volume of Linked Data sources has motivated the interest in developing applications and tools focused on consuming linked data. One of the main challenges of developing such applications is the identification of relevant sources, i.e., sources that could contribute significantly to the results of user queries submitted to the application. In this paper, we discuss this problem and present an approach to detect sources that are potentially relevant to a specific application that uses linked data. One distinguishing issue of our approach is that the process of identifying new data sources employs the user requirements expressed in SPARQL queries posed to the application.

RESUMO
O crescimento no volume de fontes de dados *Linked Data* tem despertado um grande interesse no desenvolvimento de aplicações e ferramentas voltadas para o consumo de dados interligados. Diante disso, um dos principais desafios em relação ao desenvolvimento de aplicações que utilizam dados desta natureza é a identificação de fontes relevantes, ou seja, aquelas capazes de contribuir de maneira significativa com os resultados de consultas de usuários submetidas à aplicação. Neste artigo, discutimos este problema e apresentamos uma abordagem para detectar fontes de dados que sejam potencialmente relevantes para uma determinada aplicação que consome dados interligados. Uma característica importante da abordagem proposta é que o processo de identificação de novas fontes faz uso dos requisitos de usuários expressos nas consultas SPARQL da aplicação.

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

General Terms
Algorithms, Management, Experimentation.

Keywords
Semantic Web, Linked Data, Web Crawling.

1. INTRODUÇÃO
Diversas iniciativas, como as desenvolvidas pelo W3C (*World Wide Web Consortium*), buscam, por intermédio da criação de padrões, arquiteturas de metadados, serviços de inferências e ontologias, soluções que facilitem o compartilhamento e o processamento de dados disponíveis na Web. Dentre estas iniciativas, destaca-se a Web Semântica (*Semantic Web*), uma extensão da Web atual, onde o conteúdo publicado está associado a um significado que é compreensível tanto por um humano quanto por uma máquina. No contexto da Web Semântica, o termo *Linked Data* (Dados Interligados) é utilizado para descrever um conjunto de práticas para publicação de dados estruturados na Web, de forma a aumentar o valor e a utilidade desses dados.

Com a crescente adoção dos padrões de *Linked Data*, também tem aumentado o número de aplicações que fazem uso destes dados. Estas aplicações podem ser genéricas, como os navegadores e os motores de buscas, uma vez que oferecem acesso a dados de diversos domínios, ou podem ser aplicações específicas, uma vez que oferecem acesso a domínios específicos (como o de dados bibliográficos ou governamentais, por exemplo). Um dos principais desafios no desenvolvimento das aplicações de domínio específico é a identificação de fontes de dados relevantes, ou seja, fontes de dados que poderão contribuir com informações úteis do ponto de vista do usuário da aplicação [2, 5]. Considerando um número potencialmente grande de conjuntos de dados interligados atualmente disponíveis na Web, detectar manualmente as fontes relevantes para uma aplicação pode se tornar uma tarefa inviável.

Neste trabalho, abordamos o problema da identificação de fontes de dados relevantes para aplicações de domínio específico que consomem dados interligados, ou seja, dados publicados de acordo com os princípios de *Linked Data*. A abordagem proposta faz uso dos requisitos de usuários, extraídos a partir das consultas submetidas à aplicação, a fim de guiar a busca por novas fontes de dados. Especificamente, estamos interessados em encontrar novas fontes de dados RDF a partir das informações que podem ser extraídas dos padrões de triplas presentes em um conjunto de consultas SPARQL. A utilização desta abordagem permite a detecção de fontes de dados que não foram consideradas inicialmente, mas que, potencialmente, irão contribuir com os resultados das consultas fornecidos aos usuários.

O restante do artigo é organizado como se segue. Na Seção 2, são apresentados os principais conceitos relacionados às tecnologias da Web Semântica que compõem a base para o nosso trabalho. A Seção 3 descreve a abordagem proposta para busca de fontes RDF na Web. A Seção 4 dá detalhes de implementação da abordagem proposta e de experimentos realizados. A Seção 5 apresenta trabalhos relacionados. Por fim, a Seção 6 conclui o artigo, citando contribuições deste trabalho e indicando pesquisas futuras.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão apresentados alguns conceitos preliminares e terminologias que serão utilizadas ao longo deste artigo.

2.1 URI e RDF

URI[1] é uma sequência de caracteres que identifica ou denomina 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 derreferenciamento de URIs, que consiste no acesso via HTTP a uma URI, obtendo-se um conjunto de descrições RDF [1].

RDF[2] 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*. Como exemplo, é apresentada na Figura 1 uma tripla RDF, extraída da fonte de dados *DBpedia*[3], expressando que a UFPE se localiza na cidade do Recife. O nó mais à esquerda é o *Sujeito*, um recurso que representa a Universidade Federal de Pernambuco (UFPE), conectado ao *Objeto*, um recurso que representa a cidade Recife, através de uma aresta rotulada pelo *Predicado dbpprop:city*.

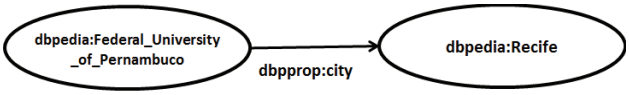

Figura 1. Exemplo de tripla RDF

2.2 SPARQL

SPARQL[4] é uma linguagem de consulta para dados RDF, permitindo a recuperação de informação contida em grafos. As principais partes de uma consulta SPARQL são [4]: 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; os *modificadores de solução*, que permitem reorganizar o resultado da consulta e a *saída*, que especifica o formato do resultado.

Como um exemplo, vamos considerar a consulta SPARQL apresentada na Figura 2, que extrai informações do *DBpedia* sobre pessoas que nasceram na cidade de Recife. Podemos identificar nesta consulta: i) o formato do resultado da consulta *SELECT ?nomePessoa*, ii) o BGP da consulta, contido na cláusula WHERE, que descreve os padrões com os quais as triplas resultantes devem estabelecer correspondência.

Q1. Retorne o nome de todas as pessoas que nasceram em Recife.

```
SELECT ?nomePessoa WHERE
{
        { ?pessoa dbp-owl:birthPlace ?cidade. }
        { ?pessoa foaf:name ?nomePessoa . }
        FILTER (?cidade = dbpedia:Recife)
}
```

Figura 2. Exemplo de consulta SPARQL

Fontes de dados interligados tipicamente fornecem um SPARQL *endpoint*[5], um serviço Web que permite ao usuário (humano ou máquina) submeter consultas SPARQL sobre os dados RDF disponibilizados na fonte.

2.3 Web Crawler Semântico

A extração de dados na Web pode ser realizada por meio de Web *crawlers*, agentes de software que acessam a Web de maneira automatizada, navegando entre os recursos por meio de links [7]. O processo realizado por este agente é chamado de *crawling*. Trabalhando sobre dados interligados, os Web *crawlers* semânticos diferem dos *crawlers* tradicionais em dois aspectos: o formato dos documentos em que navegam e o significado dos links entre as informações [6]. De maneira geral, os *crawlers* semânticos possuem a arquitetura apresentada na Figura 3 e descrita a seguir.

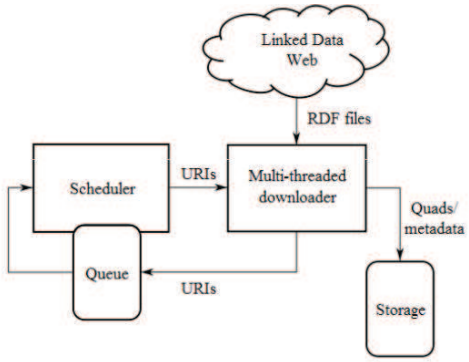

Figura 3. Arquitetura genérica de *crawlers* semânticos [7]

Um *crawler* semântico inicia sua busca de dados na Web a partir de um conjunto de recursos de origem, denominados s*eeds*, os quais são carregados em uma fila de URIs (*Queue*), onde são escalonadas uma a uma, pelo *Scheduler*, para a busca de conteúdo na Web de dados relacionados a cada recurso da fila. Cada URI escalonada é derreferenciada, obtendo-se um conjunto de triplas RDF, nas quais a URI aparece como sujeito ou objeto. A partir desta lista de triplas, são definidas novas URIs que serão inseridas na fila para próximas rodadas do *crawling*, opcionalmente pode-se restringir predicados para a obtenção de novas URIs [6].

3. A ABORDAGEM PROPOSTA

O número crescente de fontes de dados interligados disponível na Web faz surgir a necessidade de mecanismos que ajudem a filtrar tais fontes a fim de detectar aquelas que são mais relevantes de acordo com algum critério específico. Na abordagem proposta neste trabalho, aplicamos como filtro os requisitos de usuários que podem ser extraídos das consultas mais frequentemente submetidas a uma dada aplicação *Linked Data* de domínio específico.

A abordagem de busca de fontes proposta pode ser dividida em duas etapas:

(i) Extração de recursos mais relevantes de acordo com a frequência dos padrões de triplas das consultas da aplicação.

(ii) Realização de um *crawling* na Web para detectar fontes de dados interligados que descrevam os principais recursos extraídos na etapa anterior.

Estas etapas são descritas no Algoritmo *DatasetsSearch* (Algoritmo 1), que recebe como entrada um conjunto de consultas da aplicação e fornece como saída uma lista de fontes de dados candidatas que são possivelmente relevantes para esta aplicação,

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

especificamente, um conjunto de SPARQL *endpoints*. O Algoritmo 1 é detalhado nas subseções a seguir.

Algorithm DatasetsSearch
Input Q: A set of queries
k: Number of relevant resources to be considered during the crawling
Output *DE:* A set 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. $DE \leftarrow \emptyset$
7. **For each** $p \in ProvenanceTriples$ **do**
8. $DE \leftarrow DE \cup RetrieveSparqlEndpoint(p)$
9. **End for**
10. **Return** DE
End

Algoritmo 1. Algoritmo para Busca de Fontes de Dados

3.1 Extração de Recursos Relevantes

O primeiro passo do algoritmo consiste em usar a função *ExtractRelevantResources* para a identificação do conjunto de recursos relevantes, ou seja, recursos que irão guiar a busca por fontes de dados candidatas. De uma maneira geral, um recurso é identificado por uma URI e pode ser um sujeito ou objeto de um padrão de tripla. A função *ExtractRelevantResources* é apresentada no Algoritmo 2, onde podemos ver que este recebe como entrada o conjunto de consultas Q e retorna uma lista dos recursos mais frequentes de Q.

Especificamente, a extração de recursos consiste na recuperação do *BGP* para cada uma das consultas em Q e para cada padrão de tripla de um dado *BGP*, seus elementos (sujeito, predicado e objeto) são visitados. Ao longo desta visita, temos a construção de uma lista de recursos, presentes nos padrões de triplas das consultas, e de suas respectivas quantidades de ocorrência. No fim deste processo, os k recursos mais frequentes serão selecionados como sendo os mais relevantes.

3.2 Web Crawling para a Busca de Fontes

Uma vez que os recursos mais relevantes são identificados, o próximo passo consiste em realizar um processo de *crawling* na Web para buscar o conjunto de fontes candidatas. O processo de *crawling* considera como *seeds* os k primeiros recursos da lista *RR* (*RelevantResources)*, que é composta por URIs que representam recursos relevantes identificados a partir do conjunto de consultas Q. Um conjunto de predicados (*rdfs:seeAlso*, *owl:sameAs* e *owl:equivalentClass*) é utilizado durante o *crawling* para permitir a obtenção de novos recursos da Web que são similares aos recursos s*eeds*.

Ao final do *crawling*, temos um conjunto de triplas recuperadas da Web que descrevem tais recursos, as quais são armazenadas em um repositório RDF. O próximo passo do processo de busca é a construção da lista de fontes candidatas relevantes. Para esta tarefa, é extraída a proveniência das triplas coletadas pelo Web *crawler*. A informação sobre a proveniência de uma tripla é identificada por uma URI, indicando a localização do arquivo de origem da respectiva tripla. Para cada URI de proveniência, é utilizada a função *RetrieveSparqlEndpoint*, a qual é usada para a recuperação da URI do SPARQL *endpoint* das fontes de dados de procedência das triplas.

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

A URI de cada *endpoint* coletado é inserida em um conjunto que é retornado como resultado final. Este conjunto de SPARQL *enpoints* fornece à aplicação em questão o acesso a novas fontes de dados da Web. Essas fontes são potenciais candidatas a integrarem o conjunto de fontes de dados que podem ser acessados a partir da aplicação, uma vez que, possivelmente, contribuem com a melhoria dos resultados das consultas submetidas à aplicação.

4. IMPLEMENTAÇÃO E EXPERIMENTOS

A abordagem para busca de fontes de dados interligados da Web proposta neste trabalho foi implementada como módulo de uma ferramenta, denominada *RDFilter*[6], que está sendo desenvolvida como parte de uma dissertação de Mestrado. Esta ferramenta tem o objetivo de auxiliar desenvolvedores a encontrar fontes de dados relevantes durante o processo de construção de aplicações de dados interligados sobre um domínio específico [5].

O *RDFilter* envolve duas tarefas principais: i) encontrar fontes de dados que possam responder à consultas da aplicação e ii) escolher as fontes de dados mais relevantes dentre as encontradas. Para a primeira tarefa, onde temos uma seleção de potenciais fontes candidatas, é utilizada a abordagem apresentada neste artigo. Enquanto na tarefa seguinte, as fontes de dados detectadas são classificadas de acordo com o *feedback* do usuário sobre os resultados das consultas, por meio de medidas de *precision* e *recall*, como descrito em [5].

Para o desenvolvimento do módulo para busca de fontes, foram consideradas diversas ferramentas e serviços já existentes. Dentre eles, destacamos a API *Jena*[7], que permite trabalhar com a linguagem de programação Java e com tecnologias semânticas como RDF e SPARQL. Como Web *crawler* semântico, adotamos o *LDSpider*[8]. Como repositório RDF, para armazenamento de dados extraídos pelo *crawler*, foi usado o *Jena TDB*[9], e por fim, para o acesso a esses dados por meio de consultas SPARQL através de um *endpoint* local, foi utilizado o servidor RDF *Joseki*[10].

Para a avaliação da abordagem proposta, foram realizados experimentos sobre o domínio de dados bibliográficos. O objetivo dos testes executados foi o de analisar as fontes de dados encontradas, a fim de identificar se tais fontes fornecem

[6]http://code.google.com/p/rdfilter/
[7]http://jena.apache.org/
[8]http://code.google.com/p/ldspider/
[9]http://jena.apache.org/documentation/tdb/
[10]http://www.joseki.org/

informações semelhantes e/ou complementares à fonte de dados considerada como fonte base para a execução das consultas. Na Tabela 1, são apresentados dados de um dos experimentos realizados.

Tabela 1. Dados de Experimento sobre Publicações

Fonte de dados base	*DBLP RKBExplorer*[11]
# consultas	15
# recursos relevantes selecionados	5
Fontes de dados retornadas	*ACM*[12]
	CiteSeer[13]
	DBLP L3S[14]

Neste experimento, um conjunto de consultas SPARQL foi construído considerando a fonte de dados *DBLP RKBExplorer* como fonte base. Este conjunto é constituído por 15 consultas que extraem informações sobre autores, artigos, conferências, revistas científicas, entre outros. As consultas foram fornecidas como entrada para o módulo de seleção de fontes, que selecionou os 5 recursos mais frequentes e, a partir destes, realizou uma busca na Web. Como resultado final foram obtidas URIs dos SPARQL *endpoints* de três novas fontes: *ACM*, *CiteSeer* e *DBLP L3S*. A análise destas fontes mostrou que, de fato, elas armazenam dados em comum com os da fonte base, o que indica que estas fontes podem contribuir com a melhoria dos resultados das consultas da aplicação.

5. TRABALHOS RELACIONADOS

Nesta seção são apresentadas brevemente algumas pesquisas relacionadas ao nosso trabalho. Em [3] uma abordagem é descrita para identificar fontes relevantes para interligação de dados em novas fontes publicadas. Esta abordagem envolve dois principais passos: (i) busca por recursos relevantes sobre índices semânticos, utilizando palavras-chave associadas às novas fontes e (ii) filtro de fontes irrelevantes utilizando técnicas de *matching* de ontologias. Este trabalho se difere do presente, entre outros aspectos, por sua abordagem focar na detecção de fontes relevantes para interligação, ao invés da execução de consultas.

Outro trabalho relacionado é apresentado em [8], o qual propõe uma abordagem para guiar a adição de novas fontes em um sistema de integração de dados baseado em busca por palavras-chave. Este processo permite a construção de um grafo a partir das fontes e seus relacionamentos, sobre o qual é realizada uma busca que retorna os resultados mais relevantes para o usuário.

6. CONCLUSÃO

Considerando o crescente volume de dados que estão sendo publicados na Web seguindo os padrões de *Linked Data*, se torna cada vez mais difícil detectar fontes que sejam relevantes para uma dada aplicação. Portanto, ter uma solução que ajude a identificar boas fontes de dados a serem usadas como entrada para um sistema se torna crucial.

Neste artigo, apresentamos uma abordagem para buscar entre as diversas fontes de dados interligados da Web, fontes que possivelmente vão contribuir com os resultados de consultas de determinada aplicação, utilizando os requisitos dos usuários expressos nas próprias consultas como meio para se realizar este busca.

Como trabalhos futuros, gostaríamos de destacar algumas direções:

(i) Experimentos adicionais para avaliação mais precisa da abordagem: planejamento de mais experimentos tanto sob o domínio de Publicações, assim como outros domínios.

(ii) Aperfeiçoamento do processo de busca de fontes: avaliação de uso de índices semânticos [3] para detecção de fontes de dados candidatas. Também será investigado o uso de descrições VoiD[15] para a melhoria da identificação de fontes de dados relevantes.

(iii) Estudo de caso em integração de dados governamentais: integração de um módulo de busca em um sistema de recomendação de fontes de dados interligados no domínio de dados abertos do governo brasileiro.

7. AGRADECIMENTOS

A FACEPE por amparar esta pesquisa e a todos que contribuíram para o desenvolvimento e avaliação da abordagem proposta neste trabalho.

8. REFERÊNCIAS

[1] Bizer, C., Heath, T., 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.

[2] Das Sarma, A., Dong, X. L., Halevy, A. 2011. Data Integration with dependent sources. In *Proceedings of the 14th International Conference on Extending Database Technology*, EDBT/ICDT 2011, New York, NY, USA.

[3] Nikolov, A., d'Aquin, M. 2011. Identifying Relevant Sources for Data Linking using a Semantic Web Index. In *Proceedings of the Linked Data on the Web*, LDOW 2011, Hyderabad, India.

[4] Pérez, J., Arenas, M., 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).

[5] Oliveira, H. R., Tavares, A. T., 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.

[6] Tavares, A. T., Oliveira, H. R., 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.

[7] Castillo, Carlos. 2005. Effective Web Crawling. In *ACM SIGIR Forum, Vol.39*, Nova York, NY, USA.

[8] Talukdar, P. P., Ives, Z. G., Pereira, F. 2010. Automatically incorporating new sources in keyword search-based data integration, In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD 2010, Indianapolis, Indiana, USA, 2010, 387-398.

[11] http://dblp.rkbexplorer.com
[12] http://acm.rkbexplorer.com
[13] http://citeseer.rkbexplorer.com
[14] http://dblp.l3s.de/d2r
[15] http://semanticweb.org/wiki/VoID