Pós-Graduação em Ciência da Computação

"SimSPARQL: Uma Abordagem baseada em Similaridade de Consultas SPARQL para Seleção de Fontes de Dados em Federações de Dados Interligados"

Por

Lídia Fransuelly Nunes de Melo Roque

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, DEZEMBRO/2012

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

LÍDIA FRANSUELLY NUNES DE MELO ROQUE

"Simsparql: Uma Abordagem Baseada Em Similaridade De Consultas Sparql Para Seleção De Fontes De Dados Em Federações De Dados Interligados"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADORA: PROFª. DRª. BERNADETTE FARIAS LÓSCIO

RECIFE, DEZEMBRO/2012

Catalogação na fonte

Bibliotecária Jane Souto Maior, CRB4-571

Roque, Lídia Fransuelly Nunes de Melo

SimSPARQL: uma abordagem baseada em similaridade de consultas SPARQL para seleção de fontes de dados em federações de dados interligados. / Lídia Fransuelly Nunes de Melo Roque. - Recife: O Autor, 2012.

xiii, 93 folhas: il., fig., tab.

Orientador: Bernadette Farias Lóscio.

Dissertação (mestrado) - Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2012.

Inclui bibliografia, anexo e apêndice.

1. Banco de dados. 2. Web semântica. I. Lóscio, Bernadette Farias (orientador). II. Título.

025.74 CDD (23. ed.) MEI2012 – 195

Dissertação de Mestrado apresentada por **Lídia Fransuelly Nunes de Melo Roque** à Pós Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**SimSPARQL: Uma Abordagem baseada em Similaridade de Consultas SPARQL para Seleção de Fontes de Dados em Federações de Dados Interligados**" orientada pelo **Profa. Bernadette Farias Lóscio** e aprovada pela Banca Examinadora formada pelos professores:

_____

Profa. Ana Carolina Brandão Salgado
Centro de Informática / UFPE

_____

Profa. Damires Yluska de Souza Fernandes
Gerência Educacional de Informática / IFPB

_____

Profa. Bernadette Farias Lósco
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 14 de setembro de 2012.

_____

Prof. Nelson Souto Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Dedicatória

Dedico este trabalho a toda minha família, amigos e a minha orientadora, pelo total apoio durante esses anos.

AGRADECIMENTOS

A Deus, pelo Dom da vida e pela sabedoria que me impulsionou a vencer as dificuldades.

A minha família, pelo carinho, dedicação e amor prestados durante estes 24 anos de existência, especialmente e principalmente, a minha tia Ana, a minha mãe e a minha avó pelo total apoio às minhas decisões. Ao meu namorado Douglas, por sempre entender os momentos em que precisei ficar ausente.

A meu pai que, embora não mais estando fisicamente presente, estará eternamente em minhas recordações.

A minha orientadora, Profª. Drª. Bernadette Farias Lóscio, por ter aceitado me orientar mesmo ciente de todas as limitações e dificuldades que eu carregava. Nos momentos de fraqueza, era a única pessoa que me impedia de desistir. Obrigada pelos ensinamentos, apoio e amizade oferecidos em tão pouco tempo de convívio, mas que foram fundamentais para minha vida.

Ao meu amigo e colega de grupo de estudo Hélio, por ter me ajudado em todos os momentos difíceis relacionados tanto à questões do mestrado quanto à questões pessoais. Tenho consciência de que nada pagará o que tem feito por mim.

A "minha co-orientadora", Ms. Fernanda Lígia, que mesmo informalmente aceitou me ajudar no decorrer desta pesquisa. Obrigada pela dedicação e apoio oferecidos no desenvolvimento e conclusão deste trabalho.

Aos meus amigos, em especial a Rodrigo, Rivelino, Natália, Josivan, Joyce, Fabrízio, Gabriel, Elyda, Carol, Thiago Araújo, Keyla, Nielson, Rafael Ferreira e Bruno, pelo companheirismo, apego e consideração demonstrado ao longo dos anos.

Aos meus amigos do Projeto Samsung/CIn-UFPE, pelo total apoio, ensinamentos e companheirismo que me ofereceram em todos os momentos em que fiz parte da equipe. Em especial, gostaria de agradecer a equipe de Pesquisa composta por TJ, Eduardo, Pamela, Gabriela, Thun Pin, Anderson e Drª. Patrícia Tedesco, por tudo de bom que tem me feito.

“Que ninguém se engane, só se consegue a simplicidade através de muito trabalho.”(ClariceLispector)

Resumo

O patamar de informações distribuídas em documentos que possuem formato digital cresce exponencialmente na *Web,* porém de maneira desorganizada, dificultando a recuperação e a filtragem das informações. Esta *Web* é denominada *Web* de Documentos ou *Web* Sintática, onde o processo de interpretação dos conteúdos é responsabilidade dos usuários. Para facilitar a integração dos dados e solucionar os problemas oriundos da *Web* de Documentos, surgiu a ideia da *Web* Semântica. As tecnologias associadas à *Web* Semântica facilitam o processamento das informações, provendo um significado bem definido aos dados. Outro conceito importante associado à *Web* Semântica é o de *Linked Data*, que trata-se de um conjunto de melhores práticas para publicar e conectar conjuntos de dados na *Web*. Este processo resultará na transformação da *Web* de Documentos em um único espaço global de dados estruturados, ou seja, a *Web* de Dados. A *Web* de Dados permite o desenvolvimento de inúmeras aplicações e ferramentas, as quais podem ser aplicadas para domínios genéricos ou específicos. A principal contribuição deste trabalho é a proposta de uma abordagem, denominada SimSPARQL, para auxiliar a seleção de fontes de dados no contexto do processamento de consultas em ambientes de federações de dados interligados. Uma federação de dados interligados consiste de um agrupamento de várias fontes de dados distintas e interligadas. Para o desenvolvimento da SimSPARQL, utilizamos como estratégia o cálculo da similaridade entre termos presentes nas consultas SPARQL e posterior agrupamento destas consultas. Para validar a proposta, construímos cenários de testes envolvendo um conjunto de consultas SPARQL e um conjunto de fontes de dados RDF disponíveis na *Web*. A partir dos resultados obtidos, comprovamos que o processo de selecionar fontes proposto pela SimSPARQL é significativamente menos custoso em relação a abordagens anteriores, uma vez que evita o acesso à todas as fontes de dados disponíveis na federação a medida que uma nova consulta é inserida no sistema.

Palavras-chave: *Linked Data*, *Web* de Dados, Processamento de Consultas, Fontes de Dados, Seleção de Fontes de Dados

Abstract

The level of information distributed in documents having digital format on the web gradually increases in an unorganized way, making difficult to recover and to filter the information. This *Web* is called *Web* of Documents or Syntactic *Web*, where the process of interpreting the content is the responsibility of users. To facilitate the data integration and solve problems from the *Web* of Documents, emerges the idea of Semantic *Web*. The technologies associated to Semantic *Web* enable auxiliary input process information, providing a well defined meaning to data. Another important concept, associated with the Semantic *Web*, is called *Linked Data* that is a set of the best practices to publish data on the *Web*. This process will result in the transformation of the *Web* of Documents into one single global space of structured data, in other words, the *Web* of Data. The *Web* of Data allows the development of numerous applications and tools, which can be applied to generic or specific domains. The main contribution of this work is to propose an approach, called SimSPARQL to aid the source selection of data in the context of query processing in federations of *Linked Data*. A federation of linked data consists of a grouping of several different linked data sources. For the development of the SimSPARQL approach, it was adopted a strategy to calculate the similarity between terms of SPARQL queries and the subsequent grouping of such queries. In order to validate the proposal, we built test scenarios involving a set of SPARQL queries and a set of linked data sources available on the web. Through the obtained results, we found out that the proposed process of sources selection shall be less expensive than regarding previous approaches, because it avoids the access to all data sources available at the federation.

Keywords: Linked Data, Web of Data, Query Processing, Data Source, Source Selection.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

ASP - Active Server Pages;

HTML - HyperText Markup Language HTML;

HTTP - Hypertext Transfer Protocol;

JSP - JavaServer Pages;

LOD - Linking Open Data;

NS - NameSpace;

OWL – Ontology *Web* Language;

PHP - Hypertext Preprocessor;

RDF – Resource Description Framework;

RDFS – Resource Description Framework Schema;

SGML - Standard Generalized Markup Language;

SPARQL – Simple Protocol and RDF Query Language;

URI - Unified Resource Identifiers;

URL - Uniform Resource Locator;

XML - eXtensible Markup Language;

Sumário

Capítulo

1

Introdução

Este capítulo fornece uma visão geral do trabalho discorrendo sobre o contexto relacionado à pesquisa. Apresenta a justificativa e motivação do estudo, a caracterização do problema e as questões as quais este estudo busca responder, além de apresentar os objetivos gerais e específicos do trabalho, e a maneira como a dissertação está estruturada.

1.1 Justificativa e Caracterização do Problema

Um grande volume de dados está sendo publicado na *Web* de acordo com os padrões de *Linked Data*, impulsionando a transformação de uma *Web* de Documentos (a *Web* atual) para um espaço global de dados conectados denominado *Web* de Dados. *Linked Data* é um conjunto de melhores práticas para publicação e conexão de dados estruturados na *Web* que, ao serem seguidos, permitem a definição de ligações entre os dados de forma a facilitar o compartilhamento de dados e a navegação de uma fonte de dados para outra [Bizer and Health, 2011]. A adoção de tais práticas tem levado à extensão da *Web* para um espaço global de dados interligados provenientes de diversos domínios como publicações científicas, filmes, músicas, comunidades *online* entre outros [Bizer *et al.* 2009].

Devido ao crescente volume de dados publicados seguindo os princípios *Linked Data*, tem aumentado o número de aplicações que fazem uso desses dados. Estas aplicações podem ser genéricas (aplicadas a diversos domínios de conhecimento), como

browsers semânticos e motores de busca, ou mais específicas (direcionadas a um determinado domínio de interesse), como as aplicações governamentais disponíveis nos portais data.gov[1] e o data.gov.uk[2], aplicações de turismo como DBPedia Mobile[3], aplicações para biomedicina como NCBO Resource Index[4] e Diseasome Map[5], e aplicações para o meio acadêmico como Researcher Map[6], Shortipedia[7] e Semantic MediaWiki[8]. Entretanto, um ponto em comum que precisa ser considerado por estas aplicações diz respeito ao processamento de consultas sobre a *Web* de Dados.

O processamento de consultas sobre dados interligados é semelhante ao processamento de consultas em sistemas de integração de dados, uma vez que as consultas devem ser processadas sobre dados que residem em fontes de dados autônomas e distribuídas. Entretanto, o processamento de consultas na *Web* de Dados impõe alguns desafios adicionais, são eles: i) grande volume da coleção de fontes de dados; ii) dinamicidade da coleção de fontes de dados; iii) elevado grau de heterogeneidade das fontes de dados.

De maneira geral, o processamento de consultas sobre dados interligados pode ser dividido em três etapas: i) Seleção das Fontes (*Source Discovery*); ii) Classificação das Fontes (*Source Ranking*) e iii) Avaliação da Consulta (*Query Evaluation*) [Ladwig and Tran, 2010]. A etapa de Seleção das Fontes tem por objetivo selecionar fontes de dados capazes de responder a uma dada consulta e pode ocorrer de duas maneiras, a partir do conteúdo definido em uma consulta, por meio do padrão de triplas, ou durante o processamento de consultas, a partir do conteúdo mencionado nos *links* de fontes de dados recuperadas. Após a etapa de Seleção das Fontes, é realizada a etapa de Classificação das Fontes, onde as fontes de dados são classificadas de acordo com a sua relevância para a consulta em questão, isto é, verifica-se a capacidade da fonte em fornecer respostas finais satisfatórias. Por fim, na Avaliação da Consulta, será escolhida uma estratégia de avaliação de consultas, a qual poderá priorizar ou não o planejamento e otimização das consultas.

[1] http://www.data.gov/communities/node/116/apps
[2] http://data.gov.uk/apps
[3] http://wiki.dbpedia.org/DBpediaMobile
[4] http://bioportal.bioontology.org/resources
[5] http://diseasome.eu/map.html
[6] http://researchersmap.informatik.hu-berlin.de/
[7] http://shortipedia.org
[8] http://smwforum.ontoprise.com/smwforum/index.php/SMW+LDE

De fato, das três etapas que ocorrem no processamento de consultas sobre dados interligados, neste trabalho estamos interessados na etapa de Seleção das Fontes. Especificamente, propomos uma abordagem denominada SimSPARQL, baseada em técnicas de similaridade e agrupamento de consultas SPARQL, fundamentando-se na manipulação dos termos presentes nos padrões de triplas das consultas. Para realizar a similaridade, é importante definir o conceito de consulta similar, ou seja, consultas são dita similares se o conjunto de termos representativos destas consultas são similares e, possivelmente, podem ser respondidas pelo mesmo conjunto de fontes de dados.

Neste contexto, a ideia geral da SimSPARQL consiste em criar grupos de termos representativos, extraídos de consultas similares, de maneira que novas consultas possam ser respondidas pelo mesmo conjunto de fontes de dados de consultas anteriores. Assim, quando uma nova consulta for submetida ao sistema, torna-se necessário apenas identificar o grupo de consultas similares ao qual a nova consulta pertence. Considera-se que, se para estas consultas já foram identificadas as fontes de dados relevantes, ao chegar uma nova consulta e esta for similar a alguma das consultas existentes nos grupos formados, então as fontes de dados relevantes capazes de responder a nova consulta serão as mesmas.

Para possibilitar a etapa de agrupamento das consultas, proposta como solução ao problema de Seleção de Fontes de Dados fornecido pela nossa abordagem, fez-se necessário realizar o cálculo de similaridade entre consultas SPARQL e obter um valor de similaridade relativo a cada nova consulta do sistema e relativo às consultas submetidas anteriormente. Assim, analisamos o conteúdo mencionado em cada *Triple Pattern* (Padrão de Tripla) contido na consulta, extraindo e comparando tais padrões.

Com relação aos agrupamentos, consideramos que cada grupo de consultas carregará um conjunto de termos extraídos de consultas inseridas no sistema e, a partir destes grupos, será possível associar uma nova consulta. Levantado todos os requisitos necessários ao desenvolvimento de nossa abordagem, conseguimos analisar os resultados e provar a eficácia. Provamos então a eficácia da abordagem SimSPARQL, comparando-a com outra abordagem proposta na literatura, a FedX [SCHWARTE *et al.* 2011]. Portanto, constatamos que é possível selecionar fontes de dados relevantes a uma dada consulta através do cálculo de similaridade e agrupamento entre consultas, proporcionando uma redução significativa no tempo e o custo de processamento utilizado na seleção.

1.1.2 Questões de pesquisa

Para o desenvolvimento da abordagem proposta, alguns importantes questionamentos foram considerados e atendidos, os quais serão brevemente descritos a seguir.

i) Porque a Seleção de fontes é uma tarefa árdua no contexto de processamento de consultas em *Linked Data*?

A Seleção de fontes de dados na *Web* de Dados é uma tarefa árdua, pois o processamento de consultas será executado em um ambiente heterogêneo, considerando os três grandes desafios, tais como, o crescente volume das fontes de dados inseridos na *Web* de Dados, a dinamicidade existente entre as fontes de dados, a heterogeneidade e a descrição das fontes de dados e as opções de acesso.

ii) Tendo em vista que as aplicações podem ser genéricas ou de domínios específicos, para qual tipo de aplicações esta abordagem será mais adequada?

A abordagem proposta poderá ser aplicada tanto em aplicações genéricas quanto em aplicações de domínios específicos. Porém, para o estudo de caso definido nesta pesquisa de mestrado, serão consideradas apenas aplicações utilizadas em domínios específicos de interesse, em função do imenso tamanho em que se encontra a *Web* de Dados. Então, escolheu-se como domínio específico para o estudo de caso deste trabalho o domínio de dados bibliográficos, considerando um conjunto de 33 fontes de dados relacionadas à ontologia AKT (*Advanced Knowledge Techonologies Ontology*).

iii) Qual o fator decisivo para adoção de uma abordagem baseada em agrupamento de consultas para seleção de fontes?

Sabendo que a tarefa de seleção de fontes dentro do contexto de processamento de consultas não é uma tarefa trivial, pois considera fortemente fatores como volume e dinamicidade de fontes de dados, onde estas fontes compõem uma nuvem de fontes de dados, propomos como solução realizar a similaridade e agrupamento de consultas. Partimos do pressuposto de que selecionar fontes de dados descritas em um conjunto menor de fontes é significativamente menos custoso que selecionar na *Web* de dados como um todo.

1.2 Objetivos e Contribuições

Este trabalho tem como objetivo geral propor uma abordagem baseada em técnicas de similaridade e agrupamento de consultas para solucionar o problema de seleção de fontes de dados no contexto de processamento de consultas na *Web* de Dados. Como objetivos específicos deste trabalho destacamos:

a. Realizar um estudo sobre *Linked Data* e a *Web* de Dados;

b. Investigar o problema de processamento de consultas sobre dados interligados e, mais especificamente, o problema de Seleção das Fontes de dados relevantes para responder uma dada consulta;

c. Investigar técnicas para avaliar a similaridade de consultas, bem como técnicas de agrupamento de consultas;

d. Especificar uma abordagem para seleção de fontes de dados relevantes baseada em técnicas de agrupamento de consultas;

e. Implementar um protótipo e realizar experimentos para validação da abordagem proposta;

A principal contribuição deste trabalho é propor uma solução para o problema de seleção de fontes de dados em ambientes de federação de dados interligados. Além desta, é possível listar algumas contribuições:

- Especificação de uma abordagem para seleção de fontes de dados em ambientes de Federação de Dados Interligados, tendo como base a identificação de similaridade e agrupamento de consultas SPARQL.
- Definição de uma estratégia para calcular a similaridade entre consultas SPARQL e possibilitar o agrupamento dos mesmos.
- Desenvolvimento de um protótipo com funcionalidades para seleção de fontes de dados relevantes a uma consulta, possibilitando a visualização dos termos extraídos da consulta, das fontes de dados selecionadas, dos grupos de fontes de dados e dos grupos de termos formados.

1.3 Estrutura da dissertação

Esta dissertação está organizada em cinco capítulos, brevemente apresentados abaixo:

O Capítulo 1 apresenta uma contextualização do trabalho, apresentando a justificativa, a motivação e a caracterização do problema, bem como apresenta uma visão geral da abordagem proposta.

O Capítulo 2 apresenta a Fundamentação Teórica referente aos conceitos básicos necessários para o desenvolvimento desta dissertação. Neste capítulo são abordados os conteúdos referentes à *Web* de Dados, *Linked Data*, Linguagens e Tecnologias utilizadas para a *Web* Semântica como RDF e SPARQL, Processamento de Consultas na *Web* de Dados e Seleção de Fontes de Dados.

O Capítulo 3 apresenta a abordagem SimSPARQL, apresentando as etapas do processo de seleção de fontes de dados utilizada para realização da similaridade e agrupamento das consultas.

O Capítulo 4 apresenta o protótipo desenvolvido para validação do trabalho proposto, expondo suas principais funcionalidades, aspectos de implementação e resultados experimentais.

Por fim, o Capítulo 5 apresenta as considerações finais do estudo, evidenciando alguns tópicos para trabalhos futuros.

Capítulo

2

Fundamentação Teórica

Neste Capítulo apresentamos a fundamentação teórica desta dissertação, onde são apresentados os conceitos referentes aos tópicos básicos e essenciais que possibilitaram o desenvolvimento de toda a pesquisa. Tais assuntos são relacionados à *Web* de Dados e *Linked Data*, destacando tópicos sobre as linguagens RDF e SPARQL, Processamento de Consultas em *Linked Data* e técnicas para cálculo de similaridade.

2.1 *Web* de Dados e *Linked Data*

A *Web* atual ou *Web* de Documentos possibilitou aos usuários uma maior facilidade no acesso aos dados disponíveis na Internet, visto que qualquer tipo de informação pode ser armazenada e recuperada pelos usuários. Na *Web* de Documentos, o armazenamento remoto de dados é fornecido por meio de um endereço único e global, chamado URL (*Uniform Resource Locator*), capaz de codificar a localização do documento para que ele possa ser recuperado na Internet. Além disto, utiliza a linguagem HTML (*HyperText Markup Language*) para definição de documentos, os quais podem ser conectados por meio de links de hipertexto. Assim como o HTML, outras tecnologias podem ser usadas na *Web* de Documentos, tais como: o *Hypertext Transfer Protocol* (HTTP), o *PHP Hypertext Preprocessor* (PHP), *Active Server Pages* (ASP), *JavaServer Pages* (JSP), *JavaScript* e *eXtensible Markup Language* (XML).

Contudo, apesar do grande avanço que a *Web* de Documentos proporcionou à comunicação, o foco ainda é voltado para as pessoas e não para o processamento de dados automatizado. A priorização está na apresentação de informações e não no seu processamento. Nesse contexto, surgiu a necessidade de organizar e categorizar as informações disponíveis na *Web* de Documentos, de maneira a tornar as buscas mais eficientes e facilitar o processamento de dados.

Berners-Lee [2001] começou a idealizar o conceito de *Web* Semântica, afirmando que *"a Web Semântica não é uma Web separada, mas uma extensão da atual. Nela a informação é dada com um significado bem definido, permitindo melhor interação entre os computadores e pessoas*". Portanto, a *Web* Semântica busca, dentre outras coisas, facilitar a interpretação e integração dos dados na *Web*. Seu propósito é adicionar significado à informação, de forma que haja uma melhoria nas suas potencialidades.

A arquitetura da *Web* Semântica, apresentada na Figura 2.1, é dividida em camadas sobrepostas, com o objetivo de melhorar a definição dos padrões, tecnologias e/ou linguagens. Cada camada deverá ser necessariamente compatível com a camada abaixo, mas independente das camadas que estão acima.

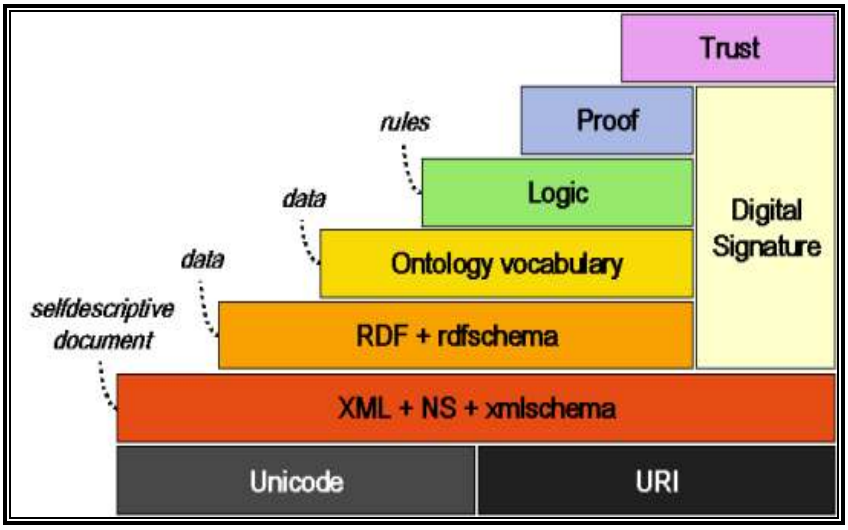


Figura 2.1: Arquitetura da Web Semântica
Fonte: (W3C, 2001)

Segundo Berners-Lee [2001], confirmando as diretrizes criadas pelo W3C [2009], as camadas da *Web* Semântica são:

- UNICODE e URI: esta camada é responsável pela interoperabilidade em relação à codificação de caracteres. O UNICODE é definido como um padrão universal de codificação de caracteres, permitindo que documentos possam ser "entendidos" por diferentes máquinas. O *Unified Resource Identifiers* (URI) é um padrão que identifica recursos. Um exemplo bastante utilizado de URI é o *Uniform Resource Locator* (URL).
- XML + NS + XMLSchema: XML é uma linguagem de marcação parecida com o *HyperText Markup Language* (HTML), ambas são derivadas da *Standard Generalized Markup Language* (SGML), porém, XML é bastante simples e mais flexível. O *Namespace* (NS) é um padrão responsável por fornecer um método para qualificar os nomes de elementos e atributos em documentos XML, preservando a individualidade dos nomes, ou seja, cada nome de elemento deve ser único. XML Schema é uma linguagem de definição de esquemas, a qual descreve uma gramática para uma classe de documentos XML, fornecendo elementos que definem a estrutura e restringem o conteúdo de documentos.
- RDF + RDFSchema: esta camada permite que seja oferecido um *framework* para representar metadados, possuindo uma sintaxe baseada em XML. *Resource Description Framework* (RDF) é um modelo para descrição de recursos na *Web*, proposto como padrão pelo W3C, ou seja, é um modelo de representação de informação na Web capaz processar metadados. RDF Schema é uma linguagem para a representação de ontologias simples, fornecendo suporte para descrição de classes e propriedades.
- *Ontology Vocabulary* (Vocabulário de Ontologia): esta camada fornece um vocabulário compartilhado acerca de um determinado domínio, modelando os conceitos, propriedades e relações do domínio em questão.
- *Logic* (Lógica): esta camada possibilita a escrita de regras especificadas em lógica de primeira ordem e permite a realização de inferências automáticas.
- *Proof* (Prova): esta camada possibilita a verificação e comprovação da coerência lógica dos recursos, onde são executadas e avaliadas as regras de inferência descritas na camada de lógica.
- *Digital Signature* (Assinatura Digital): esta camada tem como meta verificar e decidir se a informação é confiável ou não, garantindo a autenticidade do documento.
- *Trust* (Validação): esta camada visa proporcionar confiabilidade na representação das informações.

Como parte dos conceitos relacionados à *Web* Semântica, surgiu a ideia de Dados Interligados ou *Linked Data,* que pode ser definido como um conjunto de boas práticas para publicar e conectar conjuntos de dados estruturados na *Web*, com o intuito de criar uma "*Web* de Dados" [Bizer *et al*., 2009].

O conjunto de princípios *Linked Data* para publicação de dados na *Web* de Dados [Bizer *et al.,* 2009] são:

1. Usar URIs como nome para recursos.
2. Usar URIs HTTP para que as pessoas possam encontrar esses nomes.
3. Quando alguém procurar por uma URI, garantir que informações úteis possam ser obtidas por meio dessa URI, as quais devem estar representadas no formato RDF
4. Incluir *links* para outras URIs de forma que outros recursos possam ser descobertos.

Linked Data permite que todo o conteúdo da *Web* de Documentos esteja interligado, tornando-a um único espaço global denominado *Web* de Dados [Bizer and Health, 2011]. Assim, para um melhor entendimento sobre a *Web* de Dados, [Cunha *et al.,* 2011] estabeleceu um paralelo entre a *Web* de Documentos (a *Web* atual) e a *Web* de Dados. A primeira faz uso de navegadores HTML (*HyperText Markup Language*) para acessar dados na *Web* enquanto que na segunda os dados são acessados a partir de navegadores RDF *(Resource Description Framework)*. Na *Web* de Documentos *hiperlinks* são usados para navegar entre as páginas, enquanto que na *Web* de Dados os *links* RDF são usados para acessar dados de diversas fontes.

Conforme mencionado anteriormente, a *Web* de Documentos é baseada em um conjunto de padrões, incluindo: um mecanismo de identificação global e único, as URIs; um mecanismo de acesso universal, o HTTP *(HyperText Transfer Protocol*) e um formato padrão para representação de conteúdo, o HTML. De modo semelhante, a *Web* de Dados tem por base alguns padrões, como: o mesmo mecanismo de identificação e acesso universal usado na *Web* de documentos (as URIs e o HTTP); um modelo padrão para representação de dados, o RDF (modelo de dados baseado em triplas sujeito/predicado/objeto) e uma linguagem de consulta para acesso aos dados, a linguagem SPARQL.

O tema *Linked Data* traz novos desafios para o desenvolvimento de aplicações *Web* de uma maneira geral, bem como para o gerenciamento da grande nuvem de dados que vem se formando como resultado da crescente adoção destes princípios. Neste

cenário, [Ladwig and Tran, 2010] cita alguns desafios que tem sido o foco de estudo de diversos grupos de pesquisa são eles:

- Volume da Coleção de Fontes de Dados: referente ao aumento no volume de fontes de dados publicados na *Web* de Dados.
- Dinamicidade da Coleção de Fontes de Dados: referente à inserção e remoção constante de fontes de dados publicados na *Web* de Dados e às constantes mudanças no conteúdo das fontes.
- Heterogeneidade no Volume de Dados das Fontes, nas Descrições das Fontes de Dados e nas formas de acesso: referente à falta de padronização das descrições das fontes de dados presentes na *Web* de Dados, ou seja, dificuldade relacionada aos diversos tamanhos e conteúdos que as fontes de dados podem possuir.

A *Web* de Dados proporciona uma melhor organização dos dados disponíveis na *Web*, ou seja, o conteúdo das informações será considerado em cada nas páginas da *Web* e transmitido ao usuário de uma maneira mais precisa. De acordo com [Bizer and Health, 2011], a *Web* de Dados é considerada uma camada adicional, ligada aos documentos clássicos da *Web*, por meio de triplas RDF, interligando classes e propriedades de um vocabulário. Algumas características podem ser relacionadas à *Web* de Dados, tais como [Bizer and Health, 2011]:

- A *Web* de Dados é genérica e pode conter qualquer tipo de dados;
- Qualquer pessoa pode publicar dados na *Web* de Dados;
- Entidades são conectadas por *links RDF*, criando um grafo global de dados, que se estende por fontes de dados e permite a seleção de novas fontes;
- A *Web* de Dados é aberta, significando que aplicações não precisam ser implementadas sobre apenas uma fonte de dados, pois podem ser selecionadas novas fontes a partir dos *links* RDFs em tempo de execução;
- Os dados são separados rigorosamente da formatação e dos aspectos de apresentação;
- A utilização do HTTP como mecanismo padrão de acesso e do RDF como padrão de modelo de dados simplificam o acesso às fontes se comparado com as *Web APIs* as quais dependem de modelos de dados heterogêneos e interfaces de acesso disponibilizadas pelos proprietários.

O exemplo mais visível da adoção e aplicação dos princípios *Linked Data* tem sido o projeto *Linking Open Data (LOD)*[9] fundado em Janeiro de 2007 e apoiado pelo W3C *Semantic Web Education and Outreach Group*[10]. O objetivo principal deste projeto é identificar conjuntos de dados disponíveis sob licenças abertas e convertê-los para RDF de acordo com os princípios *Linked Data*.

Os participantes nas fases iniciais deste projeto foram os pesquisadores e desenvolvedores de laboratórios universitários e empresas de pequeno porte. Desde então, o projeto tem crescido consideravelmente, conseguindo um envolvimento significativo de grandes organizações como a BBC [Haunsenblas, 2009]. Este crescimento é possível graças à natureza aberta do projeto, onde qualquer um pode participar, sendo necessário apenas publicar um conjunto de dados de acordo com os princípios *Linked Data* e interligá-lo aos conjuntos de dados já existentes.

Como informação adicional, o projeto iniciou com 12 conjuntos de dados, e inclui atualmente mais de 295 conjuntos de dados diversificados e categorizados de acordo com a área de atuação, como uma junção de bilhões de triplas RDF e milhões de *links*. As Figuras 2.2 e 2.3 ilustram a evolução da nuvem de dados do LOD, onde é apresentada a primeira e a mais recente imagem da nuvem, respectivamente. É possível observar que os nós representam os conjuntos de dados dados e que as arestas representam as ligações entre os mesmos.

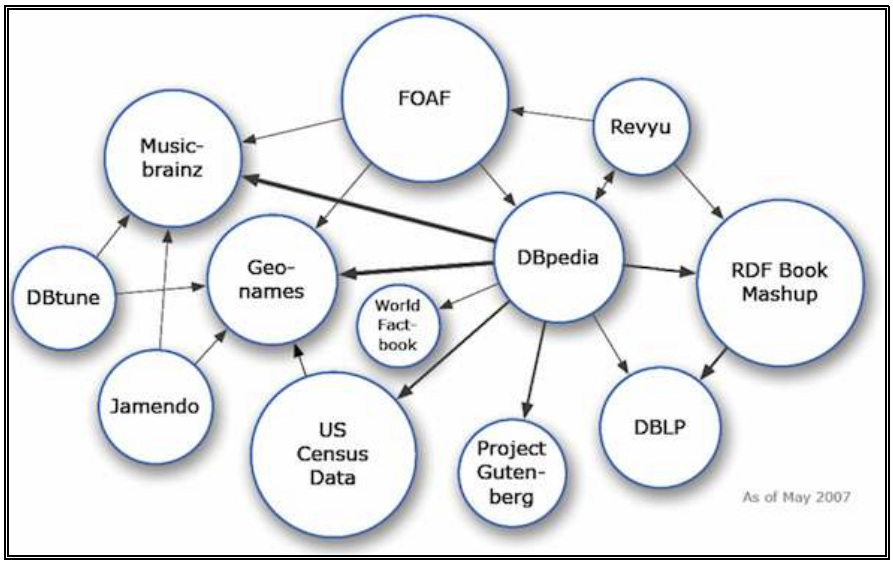


Figura 2.2: Nuvem de dados LOD em 2007
Fonte: (http://linkeddata.org/, 2007)

[9] http://linkeddata.org/
[10] http://w3c.org

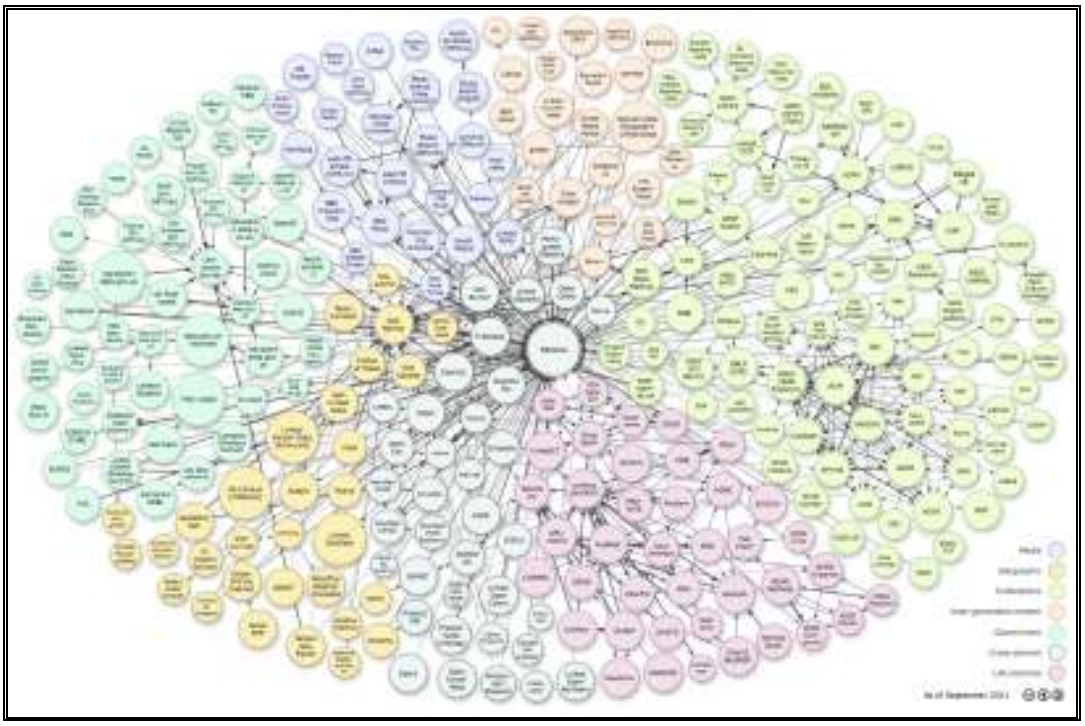


Figura 2.3: Nuvem de Dados LOD em 2011
Fonte: (http://linkeddata.org/, 2011)

2.2 RDF e SPARQL

Para possibilitar o desenvolvimento de aplicações para *Web* Semântica, alguns padrões são fornecidos na literatura e dentre eles destacamos RDF e SPARQL, os quais foram utilizados em nosso trabalho. Logo, haverá um maior detalhamento dessas linguagens.

2.2.1 Resource Description Framework - RDF

O *Resource Description Framework* (RDF) é um modelo de dados para representação de informações sobre os recursos da *Web*. Um recurso pode ser definido como qualquer objeto que possui uma identificação única chamada de URI (*Uniform Resource Identifier*) e, no contexto da *Web*, pode representar uma página *Web*, um documento eletrônico, um serviço ou uma coleção de outros recursos.

O modelo RDF descreve recursos por meio de sentenças, as quais são escritas no formato de triplas RDF. Especificamente, uma tripla RDF é constituída pela tupla <s, p, o> onde *s* é o sujeito da sentença, *p* é o predicado da sentença e *o* é o objeto da

sentença. O sujeito de uma tripla é a URI que identifica um recurso descrito. O objeto pode ser representado por um valor de literal, tal como uma *string*, um número, uma data ou pode ser uma URI de um outro recurso que está relacionada ao sujeito. Já o predicado, indica o tipo de relação que existe entre sujeito e o objeto.

Existem dois tipos distintos de triplas RDF: (i) Triplas de Literais, que possuem um literal com valor de objeto; As Triplas de Literais são usadas para descrever os atributos de recursos, como o nome ou a data de aniversário de uma pessoa. (ii) *Links* RDF, que descrevem o relacionamento entre dois recursos; Os Links RDF consistem de três referencias à URIs, onde as URIs do sujeito e objeto identificam os recurso relacionados, e as URIs do predicado definem o tipo de relacionamento entre os recursos[Bizer, 2011]. A Figura 2.4 apresenta três exemplos de triplas RDF onde a primeira e segunda linha representam Triplas de Literais e a terceira representa um *Link* RDF.

	Sujeito	**Predicado**	**Objeto**
1	http://www.w3.org/RDF/Validator/run/p91002043177	http://uni.org/uni/elements/1.1/nomeDocente	"Berna Farias"
2	http://www.w3.org/RDF/Validator/run/CK120	http://uni.org/uni/elements/1.1/nomeDisciplina	"Banco de Dados"
3	http://www.w3.org/RDF/Validator/run/CK120	http://uni.org/uni/elements/1.1/EnsinadoPor	http://www.w3.org/RDF/Validator/run/p91002043177

Figura 2.4: Exemplos de Triplas RDF

Uma tripla também pode ser representada por meio de um grafo, conforme apresentado na Figura 2.5, onde o sujeito e o objeto representam nós do grafo(URI ou um *blank node*[11]), e o predicado representa uma aresta(URI). Sendo assim, um grafo RDF pode ser constituído por um conjunto de Triplas RDF. A Figura 2.6 demonstra um exemplo de um grafo RDF contendo um conjunto de triplas RDF.

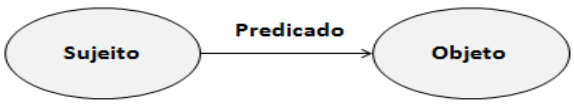


Figura 2.5: Grafo RDF

[11] *Blank Node:* nó que representa um recurso anônimo em um grafo RDF.

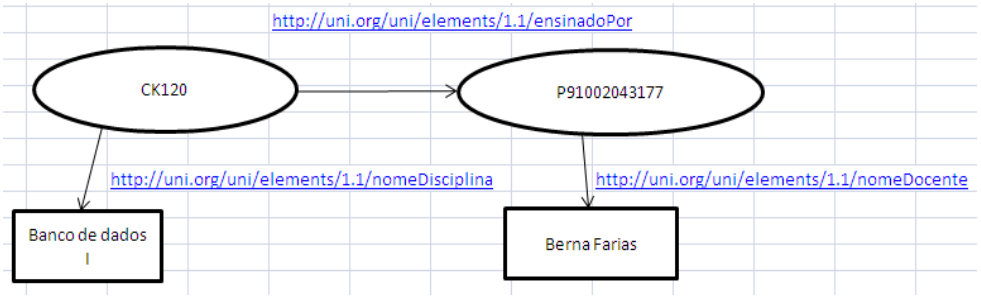


Figura 2.6: Representação Gráfica de Triplas RDF

O RDF não é um formato de dados mas sim um modelo de dados para descrever recursos no formato de triplas. Sendo assim, existem várias formas de representar os dados em RDF, tais como o RDF/XML, o RDFa, o *Turtle* e o *N-Triples*, os quais serão apresentados a seguir[Bizer, 2011].

- RDF/XML: formato padrão definido pelo W3C utilizado para publicar dados interligados na *Web*. Trata-se de uma linguagem simples e flexível, com uma sintaxe baseada em XML, que provê mecanismos de busca mais eficientes, possibilitando definições e relacionamento entre informações. Contudo, sua sintaxe é relativamente difícil para os humanos lerem e escreverem e, devido a isto, outros tipos de formatos são usados no gerenciamento de dados e fluxos de trabalhos que envolvem intervenção humana.
- RDFa: formato que associa triplas RDF em documentos HTML. Os dados RDF não são incorporados em comentários nos documentos HTML mas sim dentro de Modelos de Documentos de Objetos HTML(DOM). Isto significa que existem conteúdos nas páginas que podem ser marcados com um RDFa para modificar o código HTML, estruturando assim os dados na *Web*. O RDFa é bastante utilizado quando os editores de dados são capazes de modificar os *templates* HTML e não controlam todos os dados adicionais sobre a infraestrutura de publicação. Por exemplo, acontece quando muitos sistemas de gerenciamento de conteúdo permitem que os editores configurem templates HTML para expor diferentes tipos de informações mas não são flexíveis o bastante para suportar redirecionamento 303 e negociação de conteúdo HTTP.
- Turtle: também recomendado pelo W3C, o turtle é um formato de texto simples para serialização de dados RDF que suporta prefixos de *namespace.* Algumas informações adicionais sobre Turtle podem ser encontradas no documento fornecido pelo *W3C Team Submission* [Beckett and Bernes-Lee, 2008].

- *N-Triples*: é um subconjunto do *Turtle*, com exceção de algumas características específicas como os prefixos de *namespace*. É um formato de serialização bastante redundante, uma vez que todas as URIs devem ser completamente especificadas em cada tripla. Apesar dos arquivos *N-Triples* serem significativamente maiores que os arquivos *Turtle* ou RDF/XML, podemos considerar como vantagem o fato de que fornecem uma compreensão mais passível e reduz o tráfego na troca de arquivos em rede. Isto fez com que o *N-Triple* se tornasse padrão em *Linked Data*.

O modelo RDF impõe algumas restrições visto que não permite modelar um domínio de conhecimento, nem especificar e generalizar indivíduos, possibilitando apenas escrever sentenças sobre o domínio. Uma limitação do RDF é relacionada à impossibidade de criar vocabulários para descrever entidades e relações, e por isso é necessário utilizar o RDF Schema (RDFS). O RDF Schema é uma extensão do RDF que contém um conjunto fixo de primitivas de modelagem para construção de ontologias de domínio específico, seguindo uma ideia de hierarquia, onde o vocabulário da ontologia contém um conjunto de coleções de classes e propriedades. Assim, alguns termos utilizados pelo RDFS são descritos abaixo:

- *Class*: uma classe define um grupo de conceitos que descreve algo.
- *subClass*: uma subclasse pode ser definida como uma classe dependente de outra, seguindo uma hierarquia de classes.
- *Property*: uma propriedade é definida como um relacionamento entre classes e subclasses.
- *subPropertyOf*: uma subpropriedade representa uma hierarquia de propriedades onde uma propriedade é dependente de outra.
- *Domain*: Define quais classes de indivíduos serão aceitas como sendo o sujeito em triplas envolvendo a propriedade.
- *Range*: Define quais classes de indivíduos ou valores literais serão aceitos como sendo o objeto em triplas envolvendo a propriedade.

2.2.2 SPARQL

SPARQL é a linguagem padrão para consultar dados armazenados em grafos RDF no formato de triplas [Pérez and Arenas, 2009]. Historicamente, no ano de 2004, foi publicado o primeiro trabalho de uma linguagem de consultas para RDF, o SPARQL,

que rapidamente foi adotada como linguagem padrão para consultas semânticas na *Web* de Dados. Anos depois, em 2008, o SPARQL tornou-se recomendação do W3C para consultas sobre ontologias e grafos RDF.

A seguir, algumas definições são apresentadas para facilitar o entendimento das partes que constituem uma consulta SPARQL:

- **Endpoint:** Um endpoint SPARQL pode ser definido como um serviço que permite consultar conjuntos de dados RDF por meio da linguagem SPARQL.
- **Vocabulário:** Um vocabulário pode ser descrito como um conjunto de conceitos que podem ser usados para descrever os dados fornecidos pelos endpoints SPARQL. Várias comunidades vem criando ontologias apropriadas para descrever dados, apresentando como exemplos de vocabulários o *Friend-of-a-Friend (foaf)*, *Description-of-a-Project (doap), Semantically-Interlinked Online Communities (sioc)*.
- **Padrão de Tripla:** Corresponde aos padrões simples na forma de triplas RDF que admite a existência de variáveis em seus componentes, as quais são interpretadas como um conjunto de condições a serem retornadas no resultado da consulta.
- **Grafo RDF:** Um grafo RDF é definido como um conjunto de triplas RDF, onde o sujeito e o objeto representam os nós do grafo, e o predicado as arestas do mesmo.
- **Padrão de Grafo (GP):** Segundo [Lopes, 2009], SPARQL utiliza o casamento (*matching*) de padrões de grafo para expressar suas consultas. Devido a possibilidade de serem definidos recursivamente, é possível construir padrões complexos a partir de padrões mais simples, utilizando o conjunto de operadores algébricos fornecidos pela linguagem SPARQL, tais como: *UNION, AND, OPT, GRAPH e FILTER*. A linguagem fornece alguns padrões de grafos, tais como: o *BGP(Basic Graph Pattern), o GGP(Group Graph Pattern), o OGP(Optional Graph Pattern), o AGP(Alternative Graph Pattern) e o PNG (Patterns on Named Graphs).* Lopes [2009] e Arenas *et al.* [2009] apresentam que um Padrão de Grafo pode ser definido recursivamente como se segue:

    1. Um Padrão de Tripla também é Padrão de Grafo.
    2. Se P_1 e P_2 são padrões de grafo, então as expressões (P_1 *AND* P_2), (P_1 *OPT* P_2) e (P_1 *UNION* P_2) são padrões de grafo.
    3. Se P é um padrão de grafo e i é uma URI ou uma variável, então (*GRAPH* i P) é um padrão de grafo.

4. Se *P* é um padrão de grafo e *R* é uma expressão de filtro SPARQL, então (*P FILTER R*) é um padrão de grafo.

- **Padrão de Grafo Básico (BGP):** O conjunto de padrões de triplas utilizados em uma consulta SPARQL constitui o BGP desta consulta.

SPARQL é essencialmente uma linguagem de consultas que possui uma sintaxe similar ao SQL, porém, capaz de realizar matching entre padrões de grafos ou extrair subgrafos de uma ontologia. A estrutura da linguagem é definida em três blocos fundamentais, ilustrados na Figura 2.7 e detalhados abaixo:

(ii) Cláusulas que indicam o tipo de retorno, tais como a cláusula *SELECT* (utilizada para especificar que os resultados serão retornados para o usuário em forma de tabela), a cláusula *CONSTRUCT* (utilizada para retornar um novo grafo RDF construído através de templates, onde as suas variáveis são substituídas pelos seus valores), a cláusula *ASK* (verifica se há pelo menos uma resposta à consulta e retorna resultados booleanos como "*true*" ou "*false*", indicando se uma consulta está associada a alguma tripla) e a cláusula *DESCRIBE* (retorna um novo grafo RDF contendo recursos associados);

(iii) Cláusulas do *DATASET,* representados pela cláusula FROM*,* onde há uma especificação de quais fontes de dados RDF devem ser consultadas;

(iv) Cláusula *WHERE*, utilizada para especificar o padrão de grafo a ser buscado no conjunto de dados RDF relacionado ao *DATASET*. Apesar da cláusula *WHERE* possuir uma sintaxe inicialmente simples, este padrão pode conter alguns operadores extras como *FILTER* (utilizado para adicionar restrições aos padrões de grafos), *DISTINCT* (utilizado para remover resultados duplicados), *PROJECTION* (utilizado para reduzir a resposta ao subconjunto de variáveis definidos na cláusula *SELECT*), *REDUCED* (utilizado para permitir que resultados duplicados sejam removidos), OFFSET (indica que o resultado deve ser exibido a partir de um número de soluções determinado), OPTIONAL (utilizado para tornar uma parte do padrão opcional), *LIMIT* (utilizado para limitar o número de resultados retornados) e *ORDER BY (*ordena a saída em ordem alfabética).

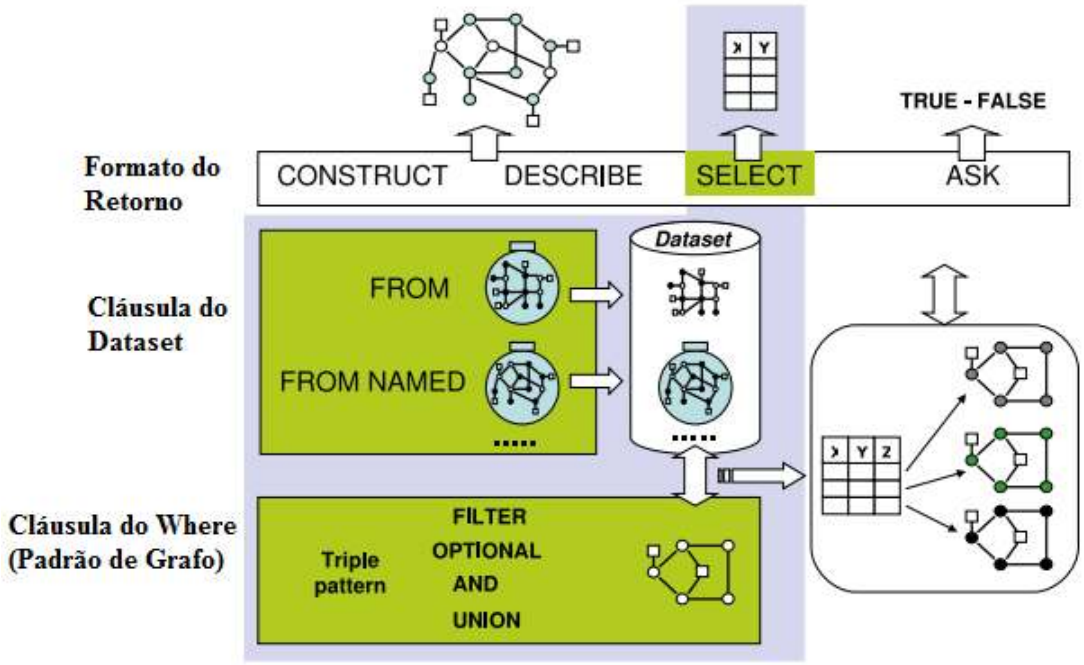


Figura 2.7: Forma Geral de uma Consulta SPARQL
Fonte: (Arenas et al, 2009)

Considere a Figura 2.8 como um exemplo de uma consulta SPARQL realizada sobre o dataset do *DotAC*[12] utilizando o vocabulário da Ontologia AKT. Esta consulta tem como objetivo recuperar informações sobre o domínio de dados bibliográficos, tais como nome, cidade e o código postal de uma determinada instituição de ensino.

```
1.  PREFIX id:      <http://dotac.rkbexplorer.com/id/>
2.  PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3.  PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
4.  PREFIX owl:     <http://www.w3.org/2002/07/owl#>
5.  PREFIX foaf:    http://xmlns.com/foaf/0.1/
6.
7.  select ?nomeInstituicao ?cidade ?codigoPostal
8.  where
9.  {
10.  id:org-anglia-ac-uk    rdfs:label        ?nomeInstituicao.
11.  id:org-anglia-ac-uk    foaf:city         ?cidade.
12.  id:org-anglia-ac-uk    foaf:postcode     ?codigoPostal.
13. }
```

Figura 2.8: Exemplo de Consultas SPARQL

[12] http://dotac.rkbexplorer.com/sparql/

A consulta pode ser explicada da seguinte forma: as linhas (1 a 5) constituem o bloco de prefixos da consulta, que especificam os *namespaces* com suas abreviações, que serão usadas no decorrer da consulta; o *SELECT* (linha 7) é responsável por indicar que a consulta terá um retorno em forma de tabela, onde neste exemplo são relacionados aos dados da universidade tais como nome, cidade e o código postal; o *WHERE*(linha 8 a 13) carrega os padrões de triplas da consulta, compostos pela tupla sujeito-predicado-objeto, as quais podem conter as URIs ou as variavéis(geralmente identificadas por iniciar pelo símbolo "?". Na Figura 2.9 é possível visualizar o retorno desta consulta, apresentado por meio das variáveis da consulta e seus respectivos valores.

Binding	Value
?nomeInstituicao	Anglia Ruskin University
?cidade	CHELMSFORD
?codigoPostal	CM1 1SQ

Figura 2.9: Resultados da Consulta SPARQL

2.3 Processamento de Consultas na *Web* de Dados

Em abordagens de processamento de consultas federadas, uma consulta está relacionada a uma federação de fontes de dados, onde esta consulta é dividida em subconsultas que poderão ser respondidas pelas fontes de dados da federação. Nestes tipos de abordagem, existe um mediador de consultas, que analisa e decompõe a consulta do usuário em várias sub-consultas. Estas sub-consultas são distribuídas em fontes de dados autonômas para serem executadas e fornecerem resultados intermediários, transmitidos em tempo real [Haase *et al*., 2010].

O processamento de consultas na *Web* de Dados pode ser considerado uma forma particular de processamento de consultas federadas, visto que os dados sob os quais as consultas devem ser processadas encontram-se em fontes de dados autônomas e distribuídas. Além disto, alguns problemas estão relacionados ao processamento de consultas sobre a *Web* de dados, e com isto, alguns pontos de pesquisa estão sendo investigados, como é o caso de como selecionar novas fontes de dados ou de como classificar se uma fonte de dados é relevante ou não a uma dada consulta.

De acordo com [Ladwig and Tran, 2010], as principais etapas do processamento de consultas sobre conjuntos de dados interligados são:

- Seleção das Fontes (*Source Discovery*): trata-se da etapa inicial do processamento de consultas, onde as fontes de dados relevantes para responder a consulta são descobertas. Ladwig and Tran [2010] demonstram três maneiras de realizar a seleção das fontes de dados, tais como: (i) a partir da definição explícita na consulta fazendo uso de uma sintaxe especial ou como parte da definição de um padrão de triplas antes da execução da consulta; (ii) O mecanismo de consulta pode manter uma lista das fontes conhecidas, de forma que esta lista pode ser construída manualmente ou compilada por meio de consultas executadas previamente e (iii) as fontes podem ser descobertas por meio da navegação entre *links* existentes nas fontes previamente recuperadas.
- Classificação das Fontes (*Source Ranking*): conforme explicado anteriormente, o volume e a dinamicidade da coleção de fontes de dados interligados tem aumentado consideravelmente e, devido a isto, torna-se inviável recuperar e processar todas estas fontes de dados. Para isto, é importante classificar as fontes de acordo com a sua relevância para responder às consultas. A classificação das fontes utiliza as descrições das fontes disponíveis, as quais podem variar na qualidade e na completude, verificando se a fonte tem informações importantes ou não [Ladwig and Tran 2010].
- Avaliação das Consultas (*Evalution Query*): a consulta pode ser avaliada de acordo com três estratégias as quais serão detalhadas abaixo:
    - Estratégia *Top-Down* de Avaliação de Consultas: Neste tipo de avaliação, assume-se que todas as descrições das fontes são avaliadas e que o plano de consulta especifica as fontes relevantes e a ordem para recuperar e processar tais fontes. Ou seja, na estratégia Top-down as fontes são previamente conhecidas, garantido melhores resultados de avaliação ao longo do grupo de dados interligados. Portanto, o foco desta estratégia é utilizar uma estrutura de captura de dados estatísticos para melhorar o planejamento e a otimização das consultas, onde a seleção da fonte é realizada de maneira *offline* e a classificação da mesma não faz parte do processo.
    - Estratégia *Bottom-Up* de Avaliação de Consultas: Ao contrário da *Top-Down*, esta estratégia não assume que as descrições das fontes são previamente elaboradas, ou seja, as consultas são avaliadas sem planejamento e sem otimização. Na estratégia bottom-up, as fontes são

descobertas durante o processamento das consultas, seguindo os links entre estas fontes. Na estratégica *bottom-up*, quatro passos são realizados: (1) recuperação as fontes de dados que são mencionadas na consulta; (2) descoberta das novas fontes de dados baseando-se nas URIs e nos links encontrados nas fontes de dados recuperadas; (3) incorporação do conteúdo das fontes descobertas na avaliação das consultas; (4) finalização do processo quando todas as fontes relevantes são encontradas. Portanto, na estratégia *Bottom-up*, a seleção e recuperação das fontes são parte integral de um processo online e pode ser aplicada quando não existir um conhecimento prévio de fontes de dados.

- Estratégia Mista de Avaliação de Consultas: Esta estratégia combina as estratégias *Bottom-Up* e *Top-Down*, assumindo que o conhecimento sobre as fontes de dados está disponível e que mais conhecimento pode ser obtido durante o processamento de consulta. Nesta estratégia, há um planejamento de consulta "*best-effort*" e, baseado neste planejamento, a consulta evolui. Além disto, durante este processo as fontes são recuperadas, novas fontes são descobertas, novos conteúdos de fontes são incorporados de forma contínua na avaliação e novas descrições de fontes são usadas para corrigir o *ranking* das fontes e a otimização. Para lidar com o volume e dinamicidade de dados na *Web* de Dados, a estratégia mista é a mais aconselhável.

Considerando que neste trabalho estamos interessados na etapa de seleção de fontes, a seguir descrevemos com mais detalhes esta etapa do processamento de consultas.

2.4 Seleção de Fontes de Dados

Pesquisas que envolvem o processamento de consultas SPARQL e o problema da seleção de fontes de dados na *Web* de dados têm ganhado destaque nos últimos anos. Uma justificativa para a necessidade de selecionar fontes de dados na *Web* de dados se dá pelo fato de que a medida que o número de fontes disponíveis cresce, aumenta também a complexidade de otimização de consultas SPARQL sobre os dados

interligados. Dessa forma, torna-se necessário reduzir o número de fontes a serem consideradas durante o processo de uma consulta.

Neste trabalho, estamos interessados na fase de seleção das fontes de dados relevantes para responder a uma dada consulta do usuário. Especificamente, propomos uma abordagem para solucionar o problema da seleção de fontes de dados em uma federação de dados interligados. Isto se deu pois ficou-se claro que a tarefa de selecionar fontes de dados relevantes para responder a uma dada consulta é um ponto fundamental no processamento de consultas da *Web* de dados.

Alguns trabalhos que envolvem o processamento de consultas na *Web* de Dados ou, especificamente, a etapa de Seleção de Fontes tem sido propostos na literatura. Maiores detalhes sobre os trabalhos serão apresentados a seguir.

Em Hose e Schenkel [2012] é proposta uma estratégia para selecionar fontes de dados que manipula cada padrão de tripla de uma consulta. O trabalho utiliza como método a extensão de operações ASK que, além de retornar operadores booleanos (*true* e *false*), recuperam um breve sumário de resultados na forma de *bloom filters. Os bloom filters* correspondem a um conjunto de elementos para vetores de bits correspondentes, que usam funções *hash*[13] para estimar qualidade e que podem combinar novos vetores. Baseados nestes *bloom filters*, é estimado o benefício de recuperar resultados de um padrão de tripla de uma fonte, sendo descartado o benefício menor ou igual a zero. Esta estratégia para seleção de fontes pode ser ajustada para recuperar resultados possíveis de um padrão de triplas em relação ao que foi estimado em um dado planejamento, ou pode minimizar o número de requisições para recuperar todo ou parte dos resultados.

Em Harth *et al.* [2010], os autores desenvolveram uma estrutura de índices de sumários para seleção de fontes de dados, fornecendo um algoritmo para responder consultas conjuntivas na *Web* de dados, explorando sumários de fontes e avaliando o sistema usando consultas geradas sintaticamente. A proposta desse trabalho é identificar fontes de dados relevantes por meio de índices a triplas RDF, fornecidos por fontes, transformando as fontes num espaço de dados númericos que indexa os dados resultantes em sumários de dados. Estes sumários são construídos porque não é possível manter todos os itens de dados em um índice, então o sumário de dados representa uma aproximação do conjunto de fontes de dados completo. A abordagem apresenta estratégias para construção de sumários de dados de um *dataset* e usa uma estrutura de

[13] Uma função hash pode ser definida como uma função que recebe um conjunto de informações como entrada e as transforma em um resultado único de saída.

índices chamada *Qtree*, desenvolvida para processamento de consultas top-k. A *QTree* é uma estrutura de árvores que consiste de nós filhos chamados *buckets* que representam os ítens das fontes de dados, onde cada *bucket* armazena o número de triplas mapeadas em coordenadas. A *QTree* é construída incrementalmente por meio da inserção de itens de dados sequenciais. Sendo assim, para determinar as fontes de dados relevantes, os autores identificaram as regiões no espaço de dados que contém todas as triplas RDF e cada padrão de tripla é convertido em um conjunto de coordenadas no espaço de dados, de modo que seja obtido um valor através do uso de funções hash para indexar e obter as coordenadas da tripla. Uma vez identificados todos os *buckets* e a região no espaço, é calculada uma cardinalidade e, baseando nesses três fatores é possível determinar o conjunto de fontes relevantes.

Em Hose *et al.* [2011] é proposto o FedX, um método para processamento e otimização de consultas que permite consultar várias fontes de dados interligados, como se os dados estivessem em um grafo RDF integrado. O FedX utiliza como estratégia de seleção de fontes de dados relevantes o envio de consultas ASK a todas as fontes de dados existentes na federação de dados. O FedX não considera o preprocessamento de metadados e utiliza um cache para guardar informações, a fim de minimizar o número de consultas ASK enviadas, o que não diminui o custo pois a medida que uma nova consulta é inserida, o processo de envio de consultas ASK é feito novamente. Logo, o FedX possibilita a integração de *endpoints* SPARQL disponíveis em uma federação, sem necessitar de processamento local, e aplica otimizações a todas as subconsultas conjuntivas, pois durante a execução da consulta, as subconsultas são geradas e avaliadas nos endpoints relevantes.

Em Quilitz and Leser [2008] é apresentado o DARQ, um sistema de consultas federadas, que utiliza a extensão do processador Jena ARQ SPARQL para gerar planos de consultas e aplicar otimizações. O DARQ contém informações sobre *endpoints* SPARQL, vocabulários de ontologias e estatísticas, e fornece um acesso transparente à consultas em vários *endpoints* distribuídos, como se consultasse em um único grafo RDF. Os autores introduzem descrições de serviços que descrevem as capacidades dos *endpoints* SPARQL e propõem um algoritmo de otimização de consultas focado na utilização de regras de reescrita de consultas. O DARQ constrói um plano de consultas baseado em custo e esforço e considera as limitações em padrões de acesso, necessitando de algumas variáveis para determinar se existe vinculo ou não à consulta. O algoritmo utilizado para encontrar as fontes de dados relevantes relaciona todos os

padrões de triplas com as fontes de dados suas correspondentes, ou seja, existe o *matching* para comparar os predicados de um padrão de tripla com o predicado definido. Portanto, como o matching é baseado apenas em predicados, o DARQ só considera consultas que contenham predicados vinculados. Os resultados da seleção de fontes são usados para construir subconsultas que serão respondidas pelas fontes de dados.

Langegger [2008] apresenta o SemWIQ, um sistema focado no compartilhamento de dados, onde fontes de dados heterogêneas são acessadas por um mediador através de *wrappers*. O SemWIQ é baseado numa arquitetura de mediadores e *wrapper's* e num processador de consultas SPARQL especializado, buscando fontes de dados relevantes baseando-se na informação das variáveis de sujeito. Sendo assim, é capaz de integrar virtualmente os dados utilizando consultas. Apesar de não manipular *endpoints* SPARQL, o SemWIQ utiliza também a extensão *Jena* SPARQL ARQ, e possui um catalógo de registros que indica onde as fontes serão consultadas e, dependendo dos dados que o *endpoint* fornece, poderá ser determinado se o vocabulário utilizado será específico a uma ontologia ou não.

Por fim, Reddy e Kumar [2012] fornecem uma abordagem para otimização de consultas SPARQL na *Web* de dados. O objetivo desta abordagem é realizar consultas aproximadas SPARQL na *Web* de Dados baseando-se no relaxamento de consultas em repositórios RDF. Para este relaxamento, fez-se necessário a utilização de medidas de similaridade em padrões de triplas das consultas SPARQL, possibilitando o compartilhamento de condições das consultas em comum e, consequentemente, permitindo o compartilhamento dos dados sem repetição.

Todas as abordagens apresentadas nestes trabalhos relacionados serviram como base para fundamentar a abordagem proposta em nosso trabalho. Especificamente, dois trabalhos contribuíram significativamente para o desenvolvimento de nossa abordagem.

(i) O trabalho proposto em Hose *et al*. [2011], que propõe o FedX, um sistema que seleciona fontes de dados relevantes através do envio de consultas ASK a todas as fontes de dados da federação, enquanto que a nossa, também envia consultas ASK às fontes, porém propõe algumas melhorias ao fornecer como estratégia a similaridade e agrupamento entre as consultas SPARQL.

(ii) O trabalho proposto por Reddy e Kumar [2012] que, apesar de não selecionar fontes de dados e ter como objetivo a realização de consultas aproximadas, nos forneceu a ideia de similaridade entre consultas SPARQL.

Todos os trabalhos relacionados citados nesta dissertação serão brevemente descritos na Tabela 2.1.

Tabela 2.1: Resumo das abordagens para Seleção de Fontes

Abordagem	**Objetivo**	**Estratégia utilizada**
Hose e Sckenkel [2012]	O trabalho propõe uma estratégia para selecionar fontes de dados baseando-se nos padrões de triplas das consultas.	Extensão de operações ASK e criação de sumário de dados
Harth *et al.* [2010]	O trabalho seleciona fontes de dados utilizando índices a triplas RDF.	Construção de Sumários de Dados e Estrutura de Índices à triplas RDF
Hose *et al.* [2011]	O trabalho propõe um método para processamento e otimização de consultas para seleção de fontes de dados.	Envio de consultas ASK a todas as fontes de dados
Quilitz and Leser [2008]	O trabalho tem como foco propor um sistema para de federação de consultas SPARQL para seleção de fontes de dados.	Comparação de Predicados presentes nos padrões de triplas RDF.
Langegger [2008]	O trabalho tem como objetivo propor um sistema para compartilhamento de dados, onde é possível acessar fontes de dados através de um mediador wrapper.	Arquitetura de mediadores wrappers que busca, informações presentes no Sujeito dos padrões de triplas das consultas
Reddy and Kumar [2012]	O trabalho tem como objetivo otimizar consultas SPARQL baseando-se no relaxamento de consultas.	Similaridade entre padrões de triplas das consultas
SimSPARQL	O objetivo do trabalho é selecionar fontes de dados em federações de dados baseando-se na similaridade entre consultas SPARQL	Envio de Consultas ASK e realização da Similaridade entre termos presentes nos padrões de triplas das consultas

2.5 Considerações Finais

Este capítulo apresentou os principais conceitos necessários para realização desta pesquisa, envolvendo os seguintes temas: *Web* Semântica, *Web* de Dados e *Linked Data*, e o Processamento de consultas em *Linked Data*, apresentando as principais tarefas do processamento e abordando melhor a etapa de Seleção de Fontes de Dados. Além disto, destacamos alguns trabalhos relacionados à nossa linha de pesquisa, apresentando alguns pontos de comparação com a nossa abordagem.

Nos próximos capítulos serão expostos todos os questionamentos referentes à SimSPARQL, apresentando detalhes sobre a concepção, implementação e validação da abordagem.

Capítulo

3

A Abordagem SimSPARQL

Este capítulo apresenta a abordagem SimSPARQL, a qual visa lidar com o problema de seleção de fontes de dados para responder uma consulta SPARQL submetida sobre uma federação de dados *Linked Data*. A Seção 3.1 apresenta uma visão geral da abordagem proposta, enquanto a Seção 3.2 exibe as definições básicas utilizadas ao longo deste capítulo. A Seção 3.3 descreve em detalhes a abordagem SimSPARQL. Finalmente, a Seção 3.4 discorre sobre as considerações finais acerca do conteúdo apresentado neste capítulo.

3.1 Visão Geral da Abordagem SimSPARQL

A abordagem que será apresentada neste capítulo, denominada SimSPARQL, visa lidar com o problema de seleção de fontes de dados relevantes, com o intuito de responder uma consulta submetida a um conjunto de fontes de dados interligados. Dizemos que uma fonte de dados é *relevante* para responder uma dada consulta se a fonte contribui para o resultado dessa consulta.

De maneira geral, podemos caracterizar o problema a ser tratado conforme descrito a seguir. Seja um conjunto F de fontes de dados interligados sobre as quais uma aplicação dispõe de pouca (ou nenhuma) informação a respeito de seu conteúdo. Suponha que essa aplicação necessite submeter uma consulta Q sobre F. Como descobrir, de maneira eficiente, quais fontes pertencentes a F são relevantes para Q?

A solução proposta tem como objetivo reduzir o esforço na busca por fontes de dados relevantes, de forma que a aplicação não necessite percorrer sempre todo o

conjunto *F*, mas somente um subconjunto de *F* possivelmente relevante. A determinação desse subconjunto baseia-se em duas ideias principais: (i) o conhecimento sobre o conteúdo das fontes pode ser obtido gradativamente, à medida que novas consultas são submetidas ao sistema e (ii) consultas similares são respondidas por conjuntos semelhantes de fontes de dados. Dessa forma, no método proposto, são criados grupos de fontes de dados, onde o critério utilizado para formação desses grupos é o tipo de informação que as fontes agrupadas podem fornecer. Para isso, cada grupo possui um conjunto de termos associados, os quais são extraídos das consultas SPARQL submetidas à aplicação e representam os termos que podem ser encontrados nas fontes de dados relacionadas ao grupo. É importante salientar que esses grupos são atualizados sempre que necessário, conforme novas consultas são enviadas ao sistema.

A Figura 3.1 exibe uma visão geral do enfoque adotado, com as principais atividades ilustradas.

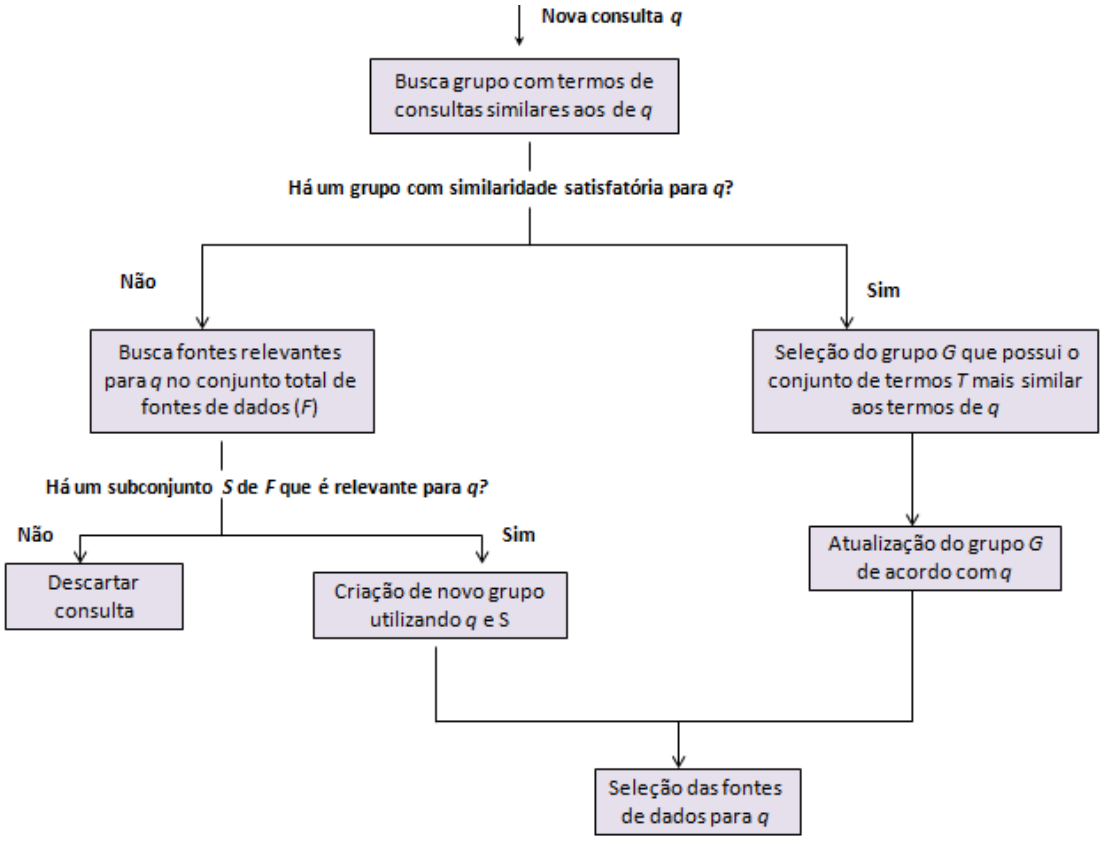


Figura3.1: Visão geral da abordagem SimSPARQL

Em linhas gerais, quando uma nova consulta *q* é submetida ao sistema, é preciso buscar o grupo com o conjunto de termos mais similares aos presentes em *q*, ou seja, o grupo no qual as fontes contenham informações sobre os termos presentes em *q*.

Caso não haja um grupo com similaridade satisfatória, é realizada uma busca no conjunto total de fontes de dados (*F*). Se alguma das fontes presentes em *F* for relevante para *q*, então um novo grupo contendo o subconjunto relevante de *F*, juntamente com os termos de *q*, é criado. Se não existir uma ou mais fontes em F relevantes para q, então a consulta não poderá ser respondida.

Por outro lado, se houver um ou mais grupos com similaridade satisfatória, o grupo de maior similaridade é selecionado. Em seguida, esse grupo é atualizado utilizando *q*, caso seja necessário. Essa atualização pode incorporar novos termos ao grupo ou sugerir a criação de novos grupos.

Finalmente, em ambos os casos em que for possível responder a consulta, as fontes associadas ao grupo em questão são utilizadas para responder à consulta *q*. As seções subsequentes descrevem de maneira detalhada cada uma das atividades mencionadas.

3.2 Conceitos Básicos

A seguir, algumas definições utilizadas ao longo deste capítulo são apresentadas, a fim de facilitar o entendimento da estratégia proposta.

Vale ressaltar que as definições 3.1, 3.2, 3.3 são adaptações das definições apresentadas no capítulo 2, de acordo com as restrições da nossa abordagem.

Definição 3.1. Padrão de Tripla (TP)

Neste trabalho, não tratamos *Blank Nodes*[14]. Portanto, definimos um padrão de tripla da seguinte forma:

Seja *U* um conjunto de URIs, *L* um conjunto de literais RDF (um literal RDF é um tipo primitivo de dados, tal como uma string, um número ou uma data) e *V* um conjunto de variáveis. Um *Padrão de Tripla (TP)* é uma tripla *(S, P, O)* ∈ *(U, V)* x *(U, V)* X *(U, L, V)*, onde *S* é o sujeito, *P* é o predicado e *O* é o objeto do padrão de tripla.

Definição 3.2. Padrão Básico de Grafo (BGP)

Um *Padrão Básico de Grafo (BGP)* é uma sequência de padrões de triplas combinados através do operador de conjunção. Neste trabalho, não estamos considerando BGPs que possuam expressões de FILTRO.

[14] Um *Blank Node* é um nó que representa um recurso anônimo em um grafo RDF.

Definição 3.3. Padrão de Grafo (GP)

Segundo Arenas *et al.* [2009], um *Padrão de Grafo (GP)* é definido recursivamente:

- Um padrão de tripla é um Padrão de Grafo.
- Outras combinações utilizando Padrões de Grafos e operadores também são Padrões de Grafos (para maiores detalhes veja o Capítulo 2 – Seção 2.2.2).

Definição 3.4. Consulta SPARQL

Neste trabalho, lidamos com consultas SPARQL nas quais o padrão de grafo (GP) é sempre um BGP. Dessa forma, uma *consulta SPARQL q* é definida pelo par $<B, T_q>$, onde:

- B representa o BGP da consulta q.
- T_q é o conjunto de termos da consulta q.

Definição 3.5. Termos de uma consulta SPARQL

Seja q uma consulta SPARQL $<B, T_q>$. Dizemos que $T_q = \{t_1,..., t_n\}$ é o conjunto de termos de q se, para cada padrão de tripla do tipo (S, P, O) existente em B, $\{S, P, O\} \subseteq T_q$. Dizemos ainda que cada $t_i \in T_q$ é chamado de *termo* de q.

Definição 3.6. Termos representativos de uma consulta SPARQL

Seja T_q o conjunto de termos de uma consulta SPARQL q. Seja V_q o conjunto de variáveis contidas em T_q. O *conjunto de termos representativos de q* é um conjunto R_q, onde $R_q = T_q - V_q$, ou seja, é o conjunto de todos os termos de q, com exceção das variáveis. Isto porque consideramos que variáveis não refletem o conteúdo existente nas fontes de dados.

Definição 3.7. Consultas SPARQL similares

Em nossa abordagem, consideramos que consultas SPARQL são similares se o conjunto de termos representativos dessas consultas são similares. De maneira mais específica, apontamos os casos descritos a seguir.

Seja q_k uma consulta SPARQL e T_k o conjunto de termos representativos de q_k. Seja ainda um conjunto de consultas SPARQL $Q = \{q_1,...,q_n\}$, onde cada q_i representa uma consulta e o conjunto de termos representativos de q_i é T_i. Diante disso, temos que o conjunto de termos representativos do conjunto de consultas Q é $T_Q = T_1 \cup ... \cup T_n$. Ou seja, T_Q contém o conjunto de termos representativos de todas as consultas q_i. Com isso, temos:

- Similaridade entre duas consultas (q_i e $q_{j'}$): q_i e $q_{j'}$ são similares se T_i e $T_{j'}$ são similares.

- Similaridade entre dois conjuntos de consultas (Q e Q'): Q e Q' são semelhantes se T_Q e $T_{Q'}$ são similares.
- Similaridade entre uma consulta (q_k) e um conjunto de consultas (Q): q_k pode ser vista como o conjunto unitário $\{q_k\}$. Assim, q_k é semelhante a Q se T_k e T_Q são similares.

Destacamos que a determinação do grau de similaridade entre os conjuntos de termos depende da fórmula adotada para calcular a similaridade entre os elementos desses conjuntos. Esta fórmula será apresentada na sessão 3.3.2 deste capítulo.

Definição 3.8. Federação de Dados Interligados.

Uma *Federação de Dados Interligados* pode ser definida como um agrupamento de várias fontes de dados RDF distintas e, possivelmente, interligadas. Sendo assim, uma federação F é um conjunto da forma $\{F_1,..., F_n\}$, onde cada F_i é um conjunto de dados RDF.

Definição 3.9. Grupo de Fontes de Dados

Dada uma federação de dados F, um *Grupo de Fontes de Dados* G é definido pelo par $<S, T_G>$, onde:

- S é um conjunto de fontes de dados associadas ao grupo G, de forma que $S \sqsubseteq F$, ou seja, S é um subconjunto de F.
- T_G é o conjunto de termos representativos do grupo G.

Definição 3.10. Termos representativos de um Grupo de Fontes de Dados

Seja G um grupo de Fontes de Dados $<S, T_G>$. Dizemos que $T_G = \{t_1,..., t_n\}$ é o conjunto de termos representativos de G se, para todo $t_i \in T_G$, temos que $t_i \in R_q$, onde R_q é o conjunto de termos representativos de uma consulta q já submetida ao sistema e para qual S é relevante. Cabe ressaltar que G é construído utilizando a ideia de similaridade entre consultas apresentada na Definição 3.7.

3.3 Abordagem SimSPARQL

Visando detalhar o funcionamento da abordagem SimSPARQL, devemos analisar dois cenários principais:

- Cenário 1 - Não há grupos de fontes de dados. Esse cenário pode ocorrer devido às seguintes situações: (i) nenhuma consulta foi submetida ao sistema ou (ii) as

consultas submetidas não foram respondidas por nenhum conjunto de fontes pertencentes à federação de dados, com isso, nenhum grupo foi criado.

- Cenário 2 - Existência de grupos de fontes de dados. Nesse caso, um ou mais grupos de fontes de dados foram criados, utilizando fontes que pertencem à federação e que são relevantes para alguma das consultas já submetidas ao sistema.

O Algoritmo 1 representa o processo geral de seleção das fontes, considerando os dois cenários mencionados anteriormente.

Algoritmo 1: ***SelecionaFontes(q,C_G, V_L)***

ENTRADA: Consulta SPARQL $\boldsymbol{q}$.
Conjunto de Grupos de Fontes de Dados $\boldsymbol{C_G}$.
Valor de similaridade mínima satisfatória (limiar) $\boldsymbol{V_L}$.
Seja *SimGrupo* o valor da similaridade entre os termos representativos T_G de $G< S,T_G >$ e T_a;

SAÍDA: Conjunto $\boldsymbol{F_q}$ de fontes de dados pertencentes à Federação de dados F e que responda q. O retorno será vazio caso não haja um subconjunto de F que responda q.

```
1.  Se C_G é vazio Então  //Cenário 1
2.      F_a ← BuscaFontesECriaNovoGrupo(q, C_G);  //Algoritmo 2
3.  Senão //Cenário 2
4.      T_q ← ExtraiTermosRepresentativos(q);  //Algoritmo 3
5.      G ← BuscaGrupoMaxSim(C_G,T_a, V_L);  //Algoritmo 4
6.      Se G é vazio Então
7.          F_a ← BuscaFontesECriaNovoGrupo(q, C_G);  //Algoritmo 2
8.      Senão
9.          Se SimGrupo = 1 Então
10.             F_a ← S de G< S, T_G >;
11.         Senão
12.             AtualizaGrupo(G, T_a);  //Algoritmo 5
13.             Se todos os termos de T_q podem ser recuperados em F Então
14.                 F_a ← S de G< S, T_G > atualizado;
15.             Senão
16.                 F_a ← Conjunto de fontes S do novo grupo criado G< S, T_G > ;
17.             Fim Se
18.         Fim Se
19.     Fim Se
20. Fim Se
21. Retorne F_q;
22.
```

Fim ***SelecionaFontes***

Este algoritmo recebe como entrada uma consulta SPARQL q, um conjunto de grupos de fontes de dados C_G e um valor de similaridade, o qual é um limiar (*threshold*) utilizado para verificar se um determinado grupo de fontes existente (pertencente a C_G) pode ser relevante para q. A saída do algoritmo é um conjunto de fontes de dados que responde à consulta q. Caso esse conjunto não exista, o retorno será um conjunto vazio.

A seguir, detalhamos cada uma das etapas executadas pelo algoritmo, o qual faz uso de outros algoritmos apresentados ao longo das explicações. Salientamos ainda que

o Algoritmo 1, por tratar do processo como um todo, será mencionado durante todo o restante deste capítulo.

3.3.1 Cenário 1 - Não há grupos de fontes de dados

Esse cenário corresponde às linhas 1 e 2 do Algoritmo 1 e está ilustrado através dos passos exibidos na Figura 3.2, os quais são descritos abaixo.

Uma vez que a consulta SPARQL *q* é recebida, inicialmente, é verificado se já existe algum grupo de consultas no sistema (Algoritmo 1 - linha 1). Caso não exista, ou seja, se o conjunto de grupos de consultas for vazio, então é preciso buscar e agrupar as fontes que são capazes de responder à consulta (Algoritmo 1 linha 2), conforme descrito a seguir.

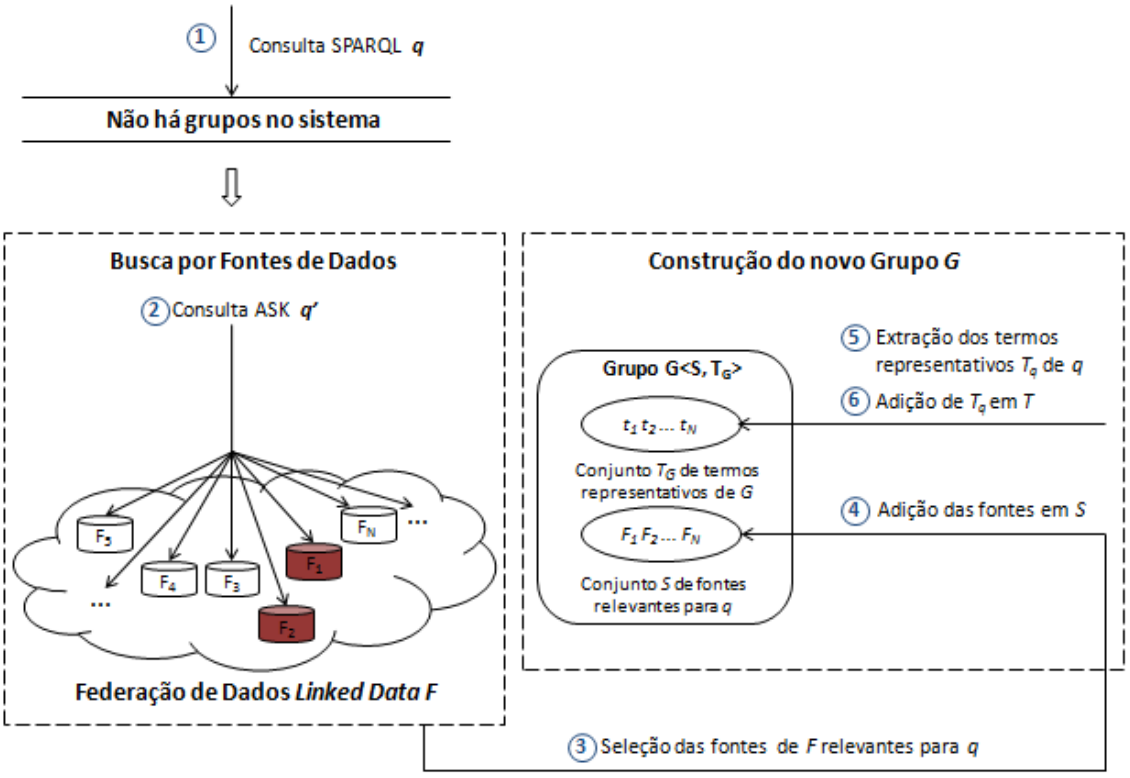


Figura3.2: Abordagem SimSPARQL para o cenário onde não há grupos de fontes no sistema

- **Busca por Fontes de Dados e Criação de Novo Grupo**

O Algoritmo 2 retrata as atividades de busca e agrupamento das fontes. Tais atividades também podem ser vistas na Figura 3.2, sendo caracterizadas pelos passos 2 a 6.

Algoritmo 2: ***BuscaFontesECriaNovoGrupo(q, C_G)***

ENTRADA: Consulta SPARQL ***q*** para qual devem ser buscadas fontes relevantes.
Conjunto de grupo de fontes de dados $\boldsymbol{C_G}$, o qual deve receber o novo grupo criado, caso existam fontes relevantes para *q*.
SAÍDA: Conjunto $\boldsymbol{F_a}$ de fontes de dados pertencentes à Federação de dados *F* e que responda *q*. O retorno será vazio se não houver um subconjunto de *F* que responda *q*.

```
1.  Fa ← ∅;
2.  Seja F o conjunto de fontes da Federação de Dados Linked Data;
3.  Fa ← EnviaConsultaASK(q,F);
4.  Se Fa não é vazio Então
5.  |  //Cria novo grupo de fontes de dados G< S, TG >
6.  |  S ← Fa;
7.  |  TG ← ExtraiTermosRepresentativos(q);  //Algoritmo 3
8.  |  CG ← CG ∪ G;  //Insere o novo grupo criado no conjunto de grupos de fontes
9.  Fim Se
10. Retorne Fa;
```

Fim ***BuscaFontesECriaNovoGrupo***

A atividade de busca por fontes de dados consiste em enviar uma consulta ASK *q'*, construída a partir de *q*, para cada uma das fontes da federação *F* (Algoritmo 2 - linha 3). A Figura 3.3 exibe um exemplo de uma consulta ASK *q'* obtida a partir de uma consulta *q*. Note que o objetivo de *q'* é apenas identificar se alguma fonte pertencente a *F* responde à *q*. Neste momento, os dados ainda não são recuperados.

Consulta SELECT *q*:

```
SELECT  ?nomeArtigo  ?nomeAutor
WHERE {
  ?artigo akt:has-title ?nomeArtigo .
  ?artigo akt:has-author ?nomeAutor }
```

Consulta ASK *q'*:

```
ASK
WHERE {
  ?artigo akt:has-title ?nomeArtigo .
  ?artigo akt:has-author ?nomeAutor }
```

Figura 3.3: Exemplo de consulta SPARQL *q* e a consulta ASK *q'* correspondente

Uma vez que as fontes são identificadas, então a criação do primeiro grupo de fontes $G_0<S_0, T_0>$ deve ser iniciada (Algoritmo 2 - linhas 4 e 5), da seguinte forma:

- As fontes selecionadas são adicionadas em G_0 (Algoritmo 2 – linha 6).
- Com intuito de definir os termos representativos de *G*, os termos representativos de *q* (T_q) são extraídos e inseridos no conjunto T_G de termos representativos do grupo *G* (Algoritmo 2 – linha 7). O Algoritmo 3 exibe como ocorre a extração dos termos representativos. Em suma, o BGP da consulta é percorrido e, para cada padrão de tripla, os termos correspondentes ao sujeito, predicado e objeto são obtidos, caso não sejam variáveis.
- Por fim, o novo grupo *G* criado é inserido no conjunto de grupos de fontes de dados do sistema (Algoritmo 2 – linha 8).

Algoritmo 3: ***ExtraiTermosRepresentativos(q)***
ENTRADA: Consulta SPARQL ***q***
SAÍDA: Conjunto T_q de termos representativos de q

```
1.  T_q ← ∅; // Conjunto de termos representativos de q
2.  Seja B o BGP de q
3.  Para cada padrão de tripla (S, P, O) ∈ B Faça
4.      //Verifica se cada termo do padrão não é uma variável e insere no conjunto de
5.      //termos representativos
6.      Se S ∉ V Então
7.          T_q ← T_q ∪ S;
8.      Fim Se
9.      Se P ∉ V Então
10.         T_q ← T_q ∪ P;
11.     Fim Se
12.     Se P ∉ V Então
13.         T_q ← T_q ∪ O;
14.     Fim Se
15. Fim Para
16. Retorne T_q;
```

Fim ***ExtraiTermosRepresentativos***

3.3.2 Cenário 2 - Existência de grupos de fontes de dados

Este segundo cenário acontece quando pelo menos um grupo de fontes de dados já foi definido. Tal cenário é retratado tanto nas linhas 3 a 20 do Algoritmo 1 quanto nas Figuras 3.4, 3.6 e 3.7.

Dada uma nova consulta q, é preciso identificar se ela pode ser respondida pelas fontes associadas a algum grupo já existente ou se há a necessidade de criação de um novo grupo. Para isso, os termos representativos de q (T_q) são extraídos por meio da chamada realizada na linha 4 do Algoritmo 1. Tal chamada, por sua vez, corresponde ao Algoritmo 3. Note que esta tarefa (bem como o algoritmo) referente à extração dos termos representativos já foi descrita na explicação do cenário anterior. Neste cenário, ela ocorre de maneira similar.

- **Busca por Grupo com maior similaridade**

Com os termos representativos da consulta q extraídos, o passo seguinte consiste em buscar o grupo que possui o conjunto de termos T_G mais similar a T_q. Essa atividade é exibida no passo 3 da Figura 3.4 e corresponde à linha 5 do Algoritmo 1.

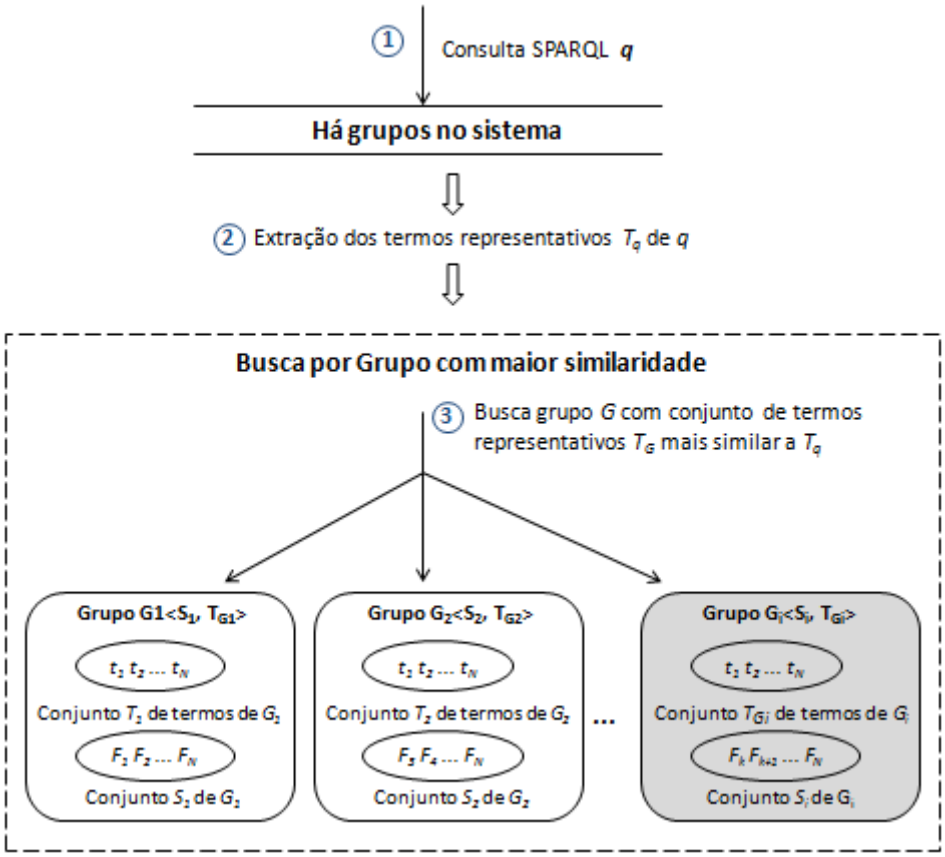


Figura 3.4: Busca por grupos de fontes de dados que respondam à consulta *q*

Conforme apresentado na Definição 3.7, consideramos que uma consulta *q* é semelhante a um conjunto de consultas *Q* se os conjuntos de termos representativos de *q* e *Q* são semelhantes. Remetemos a essa definição para utilizá-la neste cenário que está sendo analisado. De posse de *Tq*, observe que cada um dos conjuntos de termos representativos T_G associados aos grupos já existentes $G<S, T_G>$ pode ser visto como o conjunto T_Q da definição. Isso porque tais conjuntos T_G foram construídos utilizando os termos representativos de consultas submetidas anteriormente, as quais representam o conjunto *Q*.

Assim, para encontramos o conjunto de fontes que respondem a consultas semelhantes à *q*, buscamos o grupo com conjunto de termos representativos T_G mais similar a T_q. Para determinação desse grupo, é utilizado o Algoritmo 4, onde, para cada grupo de fontes $G < S, T_G >$ existente no sistema, é calculada a similaridade entre T_q e T_G. Para este cálculo é usada a fórmula apresentada na Figura 3.5.

$$\frac{min(|T_q|, |T_G|)}{|T_q \cap T_G|}$$

Figura 3.5: Fórmula para o cálculo de similaridade entre o conjunto de termos relevantes de uma consulta (T_q) e o conjunto de termos relevantes de um grupo (T_G)

Esta fórmula determina a porcentagem de termos presentes na consulta que também podem ser encontrados nas fontes associadas a G, ou seja, quantos termos de T_q estão em T_G. Observe que a fórmula é válida para ambos os casos onde a consulta possui mais ou menos termos que o grupo.

Conhecida a fórmula para calcular a similaridade entre os conjuntos de termos representativos, é possível selecionar o grupo T_G com os termos mais similares a T_q, assim como o grau de similaridade entre T_G e T_q (Algoritmo 4 – linhas 1 a 8). O grupo selecionado, que é o mais similar, precisa ainda satisfazer à seguinte condição: o valor de similaridade entre T_q e T_G deve sair maior que um limiar (*threshold*) previamente estabelecido (no Algoritmo 4, este limiar está representado por V_L). Se T_G passar nesse teste, o grupo de fontes ao qual ele está associado é selecionado (Algoritmo 4 – linhas 9 e 10). Caso contrário, o retorno será vazio, significando que no sistema não há grupo com similaridade satisfatória para q.

Algoritmo 4: *BuscaGrupoMaxSim(C_G,T_q, V_L)*

ENTRADA: Conjunto de grupo de fontes de dados $\boldsymbol{C_G}$.
Conjunto $\boldsymbol{T_q}$ de termos representativos de q.
Valor de similaridade mínima satisfatória (limiar) $\boldsymbol{V_L}$.

SAÍDA: Grupo de fontes de dados $\boldsymbol{G}$ cujo conjunto de termos representativos T_G é o mais similar a T_a. Além disso, a similaridade entre T_G e T_a deve ser maior que o limiar V_L. O retorno será vazio se não houver um grupo com T_G de similaridade satisfatória.

```
1.  Para cada grupo G < S, T_G > ∈ C_G Faça
2.      maxSim ← 0;
3.      SimGrupo ← CalculaSimilaridade(T_q, T_G);
4.      Se simGrupo > maxSim Então
5.          maxSim ← simGrupo;
6.          G_MAX ← G; //Guarda o grupo de maior similaridade
7.      Fim Se
8.  Fim Para
9.  Se maxSim > V_L Então
10.     Retorne G_MAX;
11. Senão
12.     Retorne ∅;
13.  Fim Se
```

Fim *BuscaGrupoMaxSim*

No caso de não ser selecionado nenhum grupo, o sistema irá se comportar de forma análoga ao Cenário 1 (Seção 3.4.1. Busca por Fontes de Dados e Criação de Novo Grupo), isto é, irá buscar fontes relevantes para q em toda federação F e criará novos grupos, caso uma ou mais fontes sejam encontradas (Algoritmo 1 – linhas 6 e 7).

Como descrito anteriormente, o conjunto T_G de termos do grupo selecionado deve possuir um valor de similaridade superior a um limiar. Por exemplo, um limiar de 0,8 significa que se a consulta q tiver dez (10) termos representativos ($|T_q| = 10$), então,

no mínimo, oito (8) desses termos devem estar em T_G. Note que se o limiar não for igual a 1 (100% do termos presentes em T_G), há a possibilidade de alguns termos de T_q não serem encontrados na fontes S associadas ao grupo selecionado. Diante disso, no passo 4 da Figura 3.6, é verificado se a similaridade entre os termos da consulta e os termos do grupo é 100% (Algoritmo 1 – linhas 9 e 10). Em caso afirmativo, o conjunto de fontes S do grupo selecionado G pode ser simplesmente retornado (Algoritmo 1 – linha 11). Caso contrário, é preciso atualizar o grupo G, utilizando os termos para os quais ele ainda não possui fontes associadas.

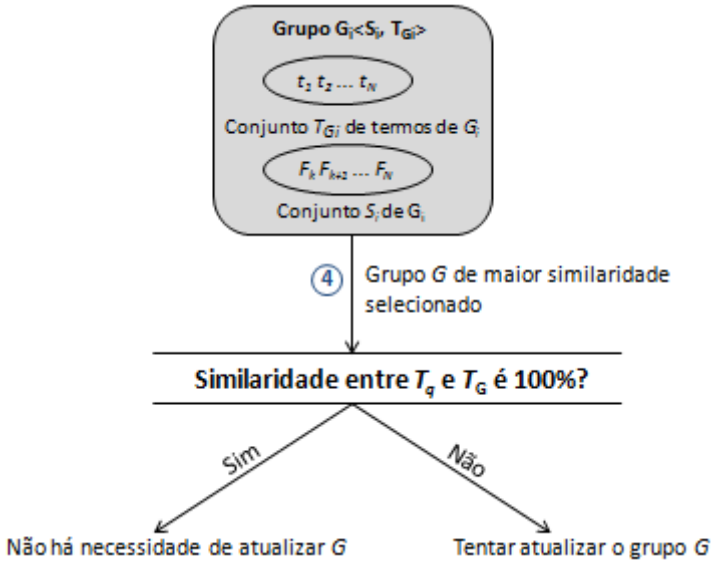


Figura 3.6: Verificação sobre a necessidade de atualização de um grupo

- **Atualização de um grupo de fontes de dados**

Essa atividade é responsável pela atualização do grupo de fontes de dados selecionado, caso seja necessário, e está representada nas linhas 13 a 18 do Algoritmo 1. Observe que o Algoritmo 1 faz uso do Algoritmo 5, pois esse último descreve o processo de atualização propriamente dito. Além disso, a Figura 3.7 também expõe as atividades realizadas.

Como já mencionado, um grupo só deve ser atualizado se algum dos termos relevantes da consulta (T_q) não estiver presente no conjunto de termos relevantes do grupo (T_G). Assim, inicialmente, são obtidos os termos presentes em T_q e que não pertencem a T_G. Chamaremos esses termos de Termos Extras (T_{EXTRAS}). Dessa forma, temos que $T_{EXTRAS} = T_q - (T_G \cap T_q)$ (Algoritmo 5 – linha 1). Não se tem informações sobre estes termos extras, de maneira que não podemos garantir que estejam (ou não)

nas fontes S associadas ao grupo selecionado. Por esse motivo, para cada termo $t \in T_{EXTRAS}$, é enviada uma consulta ASK, construída utilizando t, para o conjunto S de fontes do grupo (Algoritmo 5 – linha 4). O intuito é descobrir se as fontes já existentes possuem dados sobre t. A Figura 3.8 exibe uma consulta ASK construída para o termo *akt*:*Journal*. Como pode ser observado, essa consulta verifica se o termo está presente com as três possibilidades existentes (sujeito, predicado ou objeto) nas triplas RDF da fontes do conjunto S.

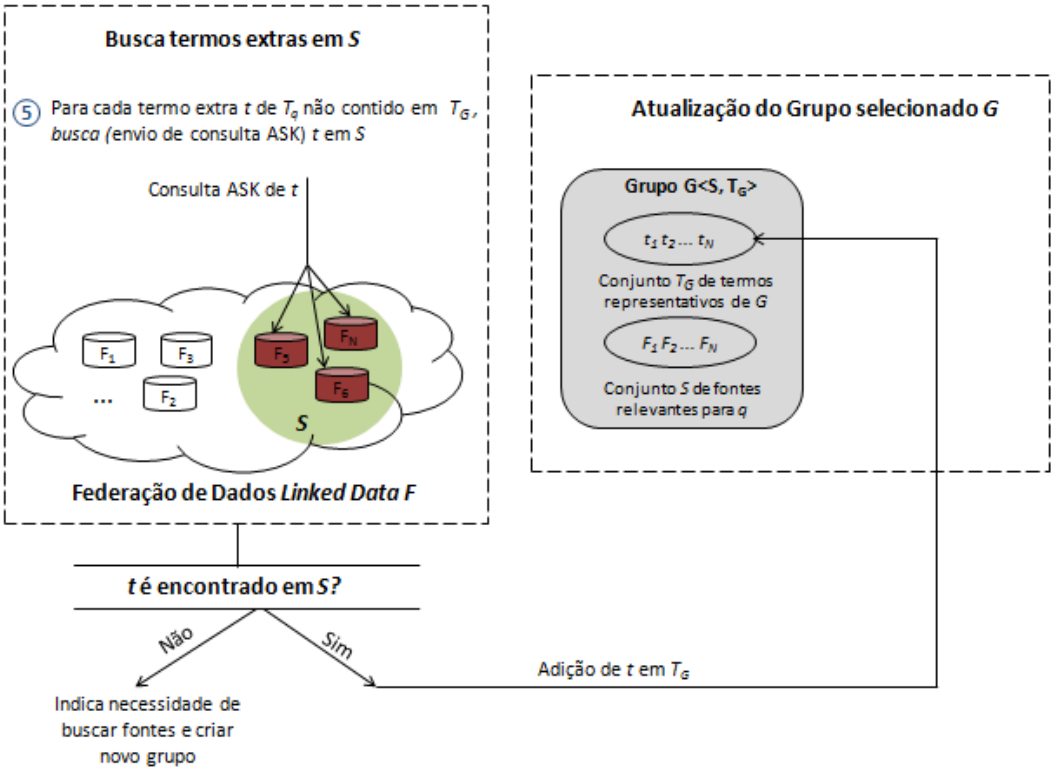


Figura 3.7: Processo de atualização do grupo G utilizando os termos extras de q

Se o termo buscado for encontrado nas fontes pertencentes ao conjunto S do grupo selecionado ($G<S, T_G>$), então o grupo deve ser atualizado. Essa atualização corresponde à adição de t ao conjunto de termos representativo T_G. Observe que o conjunto S não precisa ser atualizado, pois tal conjunto já possui as fontes que contêm t. (Algoritmo 5 – linha 7).

Termo: `akt:Journal`
Consulta SPARQL ASK do termo:

```
ASK
WHERE {
   {akt:Journal ?v1 ?v2}
   UNION
   {?v1 akt:Journal ?v2}
   UNION
   {?v1 ?v2 akt:Journal}}
```

Figura 3.8: Exemplo de consulta SPARQL ASK de um termo

Algoritmo 5: ***AtualizaGrupo(G< S, T_G >, T_q)***

ENTRADA: Grupo de fontes de dados ***G*** que pode ser atualizado
Conjunto T_q de termos relevantes de uma consulta *q*

SAÍDA: Par [Valor booleano, Grupo], onde:
- O valor booleano é uma indicação se há algum termo da consulta *q* que não pode ser encontrado no conjunto *S* de fontes do grupo mais similar selecionado. Se *Verdadeiro*, existe um termo que não pode ser recuperado, o que exigirá a possível criação de um novo grupo. Se *Falso*, todos os termos podem ser recuperados em *S*.
- Grupo *G* atualizado com novos termos, caso haja atualização ou o novo grupo G criado, caso nem todos os termos estejam em *S*.

```
1.  T_EXTRA ← T_q − (T_G ∩ T_q);
2.  // Variável que controla se algum termo de T_EXTRA não foi encontrado nas fontes
3.  naoRecuperouTermo ← falso;
4.  Para cada termo t ∈ T_EXTRA Faça
5.      // Verifica se t pode ser encontrado nas fontes do grupo G
6.      Se EnviaConsultaASKTermo(t,S) encontrou alguma fonte Então
7.          T_G ← T_G ∪ t;  // Insere t no conjunto de termos relevantes de G
8.      Senão
9.          naoRecuperouTermo ← verdadeiro; //Algum termo não foi encontrado
10.     Fim Se
11. Fim Para
12. //Verifica se há a necessidade de criação de um novo grupo para responder q
13. Se naoRecuperouTermo é falso Então
14.     Retorne [naoRecuperouTermo, grupo G< S, T_G > atualizado];
15. Senão
16.     F_a ← BuscaFontesECriaNovoGrupo(q, C_G);  //Algoritmo 2
17.     Retorne [naoRecuperouTermo, novo grupo G< S, T_G > criado];
18. Fim Se
```

Fim ***AtualizaGrupo***

Caso algum dos termos extras (T_{EXTRAS}) não possa ser encontrado nas fontes *S* do grupo *G* selecionado, então essa informação deve ser armazenada. Observe que no Algoritmo 5 (linha 9), uma variável controla tal informação pelo seguinte motivo: basta que apenas um termo da consulta *q* não seja encontrado em *S* para que o retorno da consulta inteira seja vazio, ou seja, para que *S* não responda à *q*. Isso acontece porque estamos lidando apenas com consultas cujo operador é AND (veja a Definição 3.4).

Ao final, é verificado se existe algum termo que não pode ser recuperado em *S* (Algoritmo 5 – linha 12). Em caso afirmativo, devem ser buscadas fontes de dados na federação F que sejam capazes de responder à consulta *q* e um novo grupo é, possivelmente, criado. Este novo grupo poderá ser bem similar ao grupo já existente se apenas poucos termos não forem encontrados no grupo. Esta atividade de busca e criação ocorre de maneira análoga ao explanado no Cenário 1 (Seção 3.4.1. Busca por Fontes de Dados e Criação de Novo Grupo).

É importante destacar que mesmo no caso de um termo não ser encontrado, todos os termos restantes ainda devem ser buscados antes do novo grupo ser criado. Isso

necessita ser executado para que o conjunto de termos T_G do grupo selecionado $G<S, T_G>$ seja atualizado corretamente, garantindo, assim, a manutenção e evolução do sistema para consultas posteriores.

O Algoritmo 5 retorna a informação sobre a existência (ou não) de algum termo que não possa ser recuperado nas fontes da federação *F*, juntamente com o grupo *G* atualizado (linha 13) ou o novo grupo criado (linha 16). No Algoritmo 1, nas linhas 13 a 17, essa questão é tratada.

Finalmente, na linha 22 do Algoritmo 1, o subconjunto da federação *F* que responde à consulta *q* é retornado. Note que, ao longo da execução desse algoritmo, em todas as situações onde não se encontrou um conjunto de fontes que pudesse responder à *q*, o valor vazio foi atribuído ao retorno.

3.3 Considerações Finais

A abordagem apresentada neste capítulo busca solucionar o problema de seleção de fontes de dados em ambientes de federação de dados interligados, utilizando como estratégia a identificação de similaridade entre consultas e o agrupamento de fontes de dados. Nosso objetivo consiste em evitar que todas as fontes de dados da federação sejam consultadas no momento em que uma nova consulta seja submetida ao sistema. Uma vez identificado o grupo de fontes de dados que possua maior similaridade com a consulta em questão, fontes de dados serão selecionadas em um tempo significativamente menor do que se todas as fontes fossem verificadas.

Este capítulo apresentou os algoritmos, cenários e exemplos de aplicação da abordagem. No próximo capítulo, discutimos os experimentos realizados para a validação da nossa abordagem, bem como os resultados obtidos.

Capítulo

4

Implementação e Experimentos

Este capítulo tem como objetivo abordar as questões de implementação deste trabalho, detalhando os aspectos de desenvolvimento e indicando os resultados alcançados durante a pesquisa. Para validação da abordagem SimSPARQL, foi desenvolvida uma ferramenta de mesmo nome, que é capaz de selecionar fontes de dados em federações de dados interligados. O restante deste capítulo está organizado como se segue. A Seção 4.1 apresenta a ferramenta SimSPARQL, demonstrando suas principais características e funcionalidades. A Seção 4.2 aborda as questões referentes à validação do trabalho.

4.1 A Ferramenta SimSPARQL

Com o objetivo de implementar e testar a abordagem proposta, desenvolvemos a ferramenta SimSPARQL, a qual permite que o usuário, por meio de uma interface amigável, insira uma consulta SPARQL no sistema e verifique quais fontes de dados são capazes de responder tal consulta.

4.1.1 Arquitetura da SimSPARQL

A ferramenta SimSPARQL possui uma arquitetura dividida em duas camadas: a Camada de Aplicação e a Camada de Dados. A Figura 4.1 exibe uma visão geral desta arquitetura, seguida de uma breve explanação sobre seus módulos.

- Camada de Aplicação: camada responsável por todo o funcionamento da ferramenta, agrupando o conjunto de módulos descritos abaixo. Com exceção da GUI, todos os módulos presentes nesta camada funcionam de acordo com o valor de similaridade obtido entre os termos de uma determinada consulta e os termos armazenados nos Grupos de Termos armazenados no sistema.
    - GUI (Graphical User Interface): módulo responsável por fornecer ao usuário uma interface amigável, contendo todas as funcionalidades do sistema. Assim, a GUI possibilita que o usuário interaja com todas as funcionalidades do sistema, permitindo que uma nova consulta seja inserida e, após a inserção, transfere esta consulta ao módulo Gerenciador de Grupos.
    - Extrator de Termos: módulo responsável por gerenciar os termos extraídos da consulta SPARQL T_q, agrupando-os em um conjunto de termos T_g. O conjunto de termos T_g é encaminhado ao Módulo Gerenciador de Grupos.
    - Gerenciador de Grupos: módulo responsável por identificar a qual grupo de termos T_g o conjunto T_q está relacionado, utilizando como forma de identificação, o maior valor de similaridade obtido entre os termos do conjunto T_q da consulta e os termos representativos de cada grupo T_g. Uma vez identificado qual grupo T_g é o mais representativo, o gerenciador de grupos é capaz de identificar quais fontes de dados relevantes são capazes de responder esta consulta. Além disto, este módulo é responsável por todas as tarefas relacionadas aos grupos, incluindo a criação de novos grupos e atualização de grupos existentes.
    - Repositório de Consultas: módulo responsável por associar as consultas inseridas no sistema ao grupo de fontes de dados que possui informações capazes de respondê-las. Ou seja, as consultas são associadas aos grupos correspondetes mas não são armazenadas no sistema.
- Camada de Dados: nesta camada estão armazenadas as fontes de dados participantes da federação de dados interligados. Neste trabalho, todas as fontes de dados estão relacionadas à mesma ontologia de domínio. Desta forma, foi possível usar o mesmo vocabulário na construção de todas as consultas usadas nos experimentos.

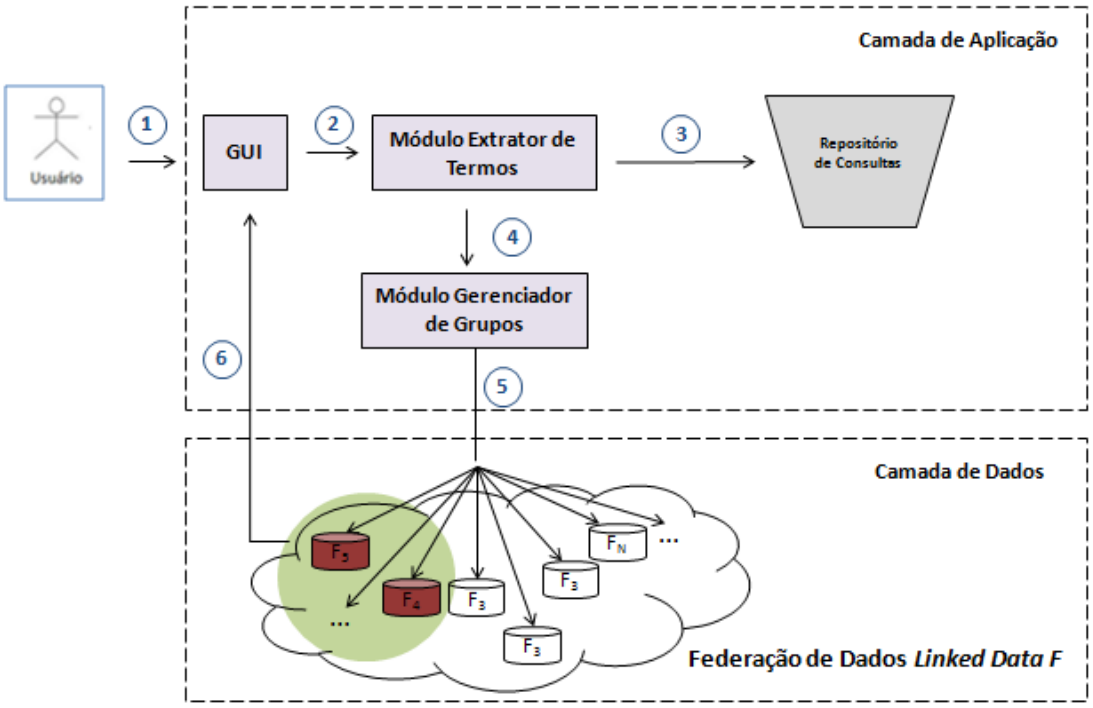


Figura 4.1: Arquitetura proposta pela SimSPARQL

Em resumo, podemos explicar a Figura 4.1 como uma sequência de passos enumerados, descritos como:

- Passo 1: A consulta é inserida na interface gráfica da ferramenta;
- Passo 2: Os termos são extraídos da consulta;
- Passo 3: Os termos são armazenados do repositório de consultas;
- Passo 4: Os termos são associados aos grupos de termos;
- Passo 5: Os termos são associados ao grupo de fontes de dados correspondentes;
- Passo 6: As fontes de dados selecionadas são apresentadas na interface gráfica da ferramenta.

A ferramenta apresentada foi implementada na linguagem JAVA, utilizando a API Jena para questões que envolveram a ontologia de domínio. Todas as consultas utilizadas no trabalho foram formuladas na linguagem SPARQL, considerando o domínio da ontologia AKT. Além disto, utilizamos o SGBD PostgreSQL 9.0 para armazenar as consultas, os termos extraídos de cada consulta e os grupos de termos representativos.

4.1.2 Especificações da SimSPARQL

Para implementação e validação da SimSPARQL, algumas tarefas foram realizadas, incluindo:

- Definição de uma técnica para similaridade de consultas.
- Definição de uma medida de similaridade entre consultas baseando-se nos termos extraídos das consultas.
- Definição de uma técnica para agrupamento de fontes de dados baseando-se nos termos das consultas.
- Definição do processo de inicialização da abordagem, ou seja, como serão definidas as consultas e os agrupamentos iniciais.
- Definição de uma estratégia para gerenciamento dos grupos de termos, uma vez que as fontes de dados e o conjunto de consultas evoluem consideravelmente.
- Definição de uma estratégia para captura de fontes de dados relacionadas a um grupo específico de termos de consultas.
- Definição e especificação de um estudo de caso, com o intuito de validar a abordagem proposta.

4.2 Funcionalidades da SimSPARQL

A Figura 4.2 apresenta o diagrama de casos de uso correspondente às funcionalidades da SimSPARQL. Serão brevemente explicadas as atividades associadas com cada funcionalidade, uma vez que o processo de seleção de fontes de dados foi apresentado no capítulo anterior.

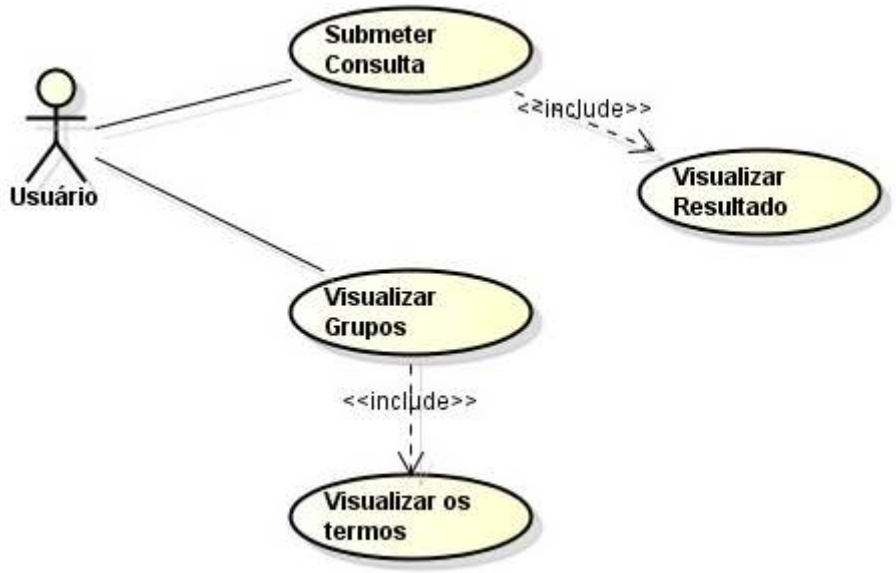


Figura 4.2: Funcionalidades da SimSPARQL

- **Submeter Consulta**

Permite que o usuário submeta uma consulta SPARQL como entrada no sistema e visualize como resultado o conjunto de fontes de dados capazes de responder tal consulta. Para execução desta funcionalidade, algumas atividades são executadas pelo sistema, as quais serão listadas a seguir.

- Extração dos Termos das Consultas: constitui a atividade inicial do sistema, realizada antes da execução da consulta, uma vez que o conjunto de termos extraídos é essencial ao desenvolvimento da abordagem proposta pela SimSPARQL.
- Envio de consultas *ASK*: esta atividade constitui a primeira parte de execução da consulta e tem como objetivo enviar consultas *ASK* a um conjunto de fontes de dados definidos no sistema. O objetivo é descobrir, dentre todas as fontes de dados cadastradas, quais as que contêm alguma informação relevante à consulta. Esta atividade poderá ser chamada no sistema em dois momentos: (i) Considerando todos os termos presentes na consulta: quando não existe nenhuma consulta anterior no sistema ou quando já foram realizadas consultas no sistema mas o valor de similaridade está abaixo do limiar configurado; (ii) Considerando termos específicos da consulta: quando o valor de similaridade está acima do limiar mas algum termo da consulta é novo ou quando o termo não pode ser respondido pelas fontes de dados presentes nos grupos.

- Criação/Seleção de Grupos: esta atividade é responsável pela realização do cálculo de similaridade entre os termos de uma consulta e os termos representativos de cada grupo. Assim, tanto é possível criar um novo grupo como escolher um grupo presente no sistema, tendo em vista que um grupo armazena um conjunto de termos específicos e o conjunto de fontes de dados relacionados a estes termos.

- **Visualizar Resultado**

Esta funcionalidade permite que o usuário visualize os detalhes da consulta inserida no sistema como: o status da consulta, ou seja, se já foi inserida ou não no sistema; o valor de similaridade da relação Consulta x Grupo; o grupo ao qual a consulta está associada; e as fontes de dados relacionadas a este grupo.

- **Visualizar Grupos e Termos**

Esta funcionalidade permite que os usuários visualizem os grupos de fontes criados e, consequentemente, a lista de termos associados a cada grupo.

É possível visualizar na Figura 4.3 a tela principal da ferramenta SimSPARQL. Nesta tela, a consulta é submetida e após sua execução, podemos visualizar informações como: grupo de termos da consulta, o valor e grupo de maior similaridade relacionado à consulta, e as fontes de dados selecionadas.

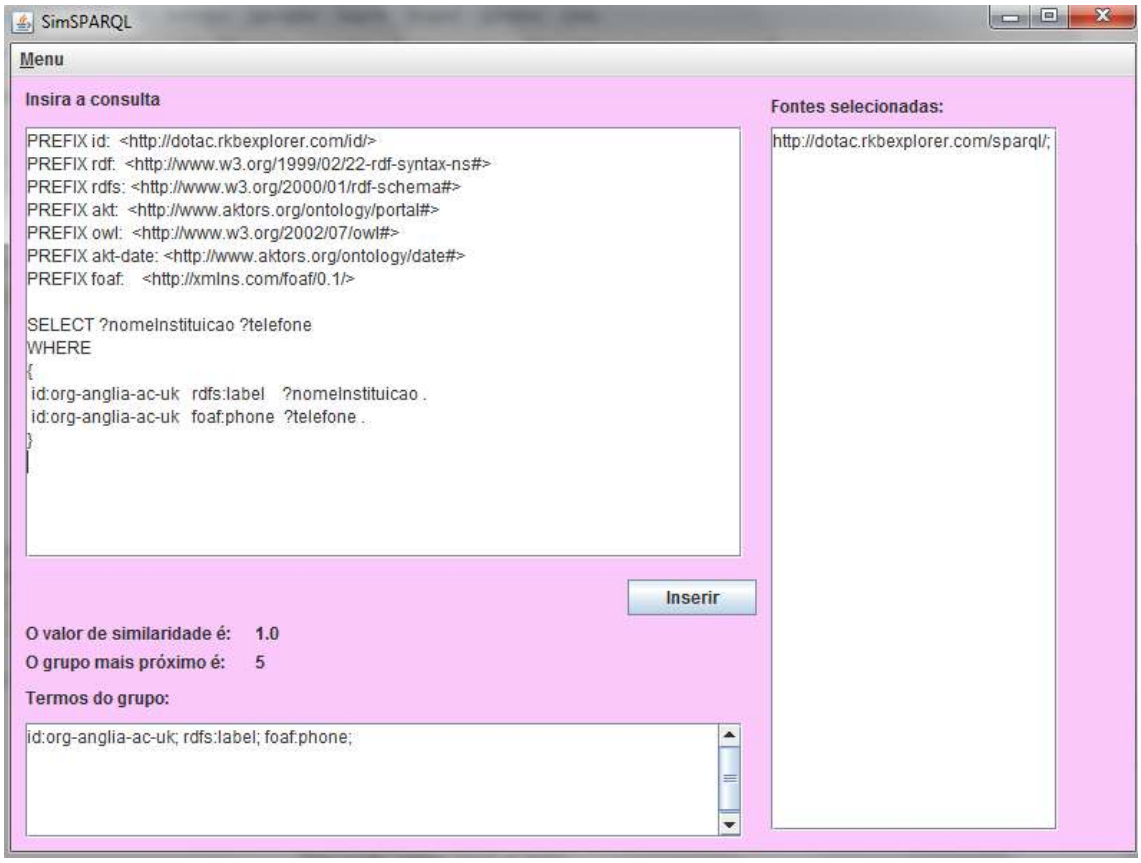


Figura 4.3: Tela Principal da SimSPARQL

4.3 Validação Experimental

Para a validação da estratégia proposta, foram realizados experimentos utilizando um conjunto de consultas e um conjunto de fontes de dados, ambos tendo como referência a ontologia AKT(*Advanced Knowledge Techonologies*[15]). Apesar de utilizarmos em nosso estudo de caso apenas consultas sobre o vocabulário fornecido pela ontologia da AKT, nossa abordagem não se limita a apenas uma ontologia e pode ser aplicada em um conjunto de ontologias, visto que analisamos os termos extraídos dos padrões de triplas.

Os experimentos foram desenvolvidos em uma máquina Sony Vaio com Sistema Operacional Windows 7, processador Core i3 e 4GB de memória. Para esta avaliação, montamos um cenário que envolve:

- Uma ontologia sobre o domínio de dados bibliográficos, representada pela ontologia *portal.owl* da AKT; composta por 264 classes e 108 propriedades. Parte da ontologia da AKT poderá ser visualizada no Anexo A deste trabalho.

[15] http://www.aktors.org/akt/

- Um conjunto com 30 consultas geradas a partir do vocabulário definido pela ontologia AKT, de acordo com alguns critérios: são consideradas apenas consultas com BGP, entretanto são desconsiderados BGPs que possuem expressões de filtros; a seleção das consultas foi realizada considerando tanto consultas semanticamente similares quanto distintas. Este conjunto de consultas encontram-se no Apêndice A deste trabalho, apresentadas tanto em suas formas originais (linguagem natural) quanto na linguagem SPARQL.
- Um conjunto com 33 *datasets*, cada qual com seu respectivo *endpoint* SPARQL, sobre os quais as consultas podem ser submetidas. Esta lista de fontes de dados pode ser encontrada no Anexo B desta dissertação. Entretanto, vale ressaltar que não podemos garantir que todas as fontes de dados desta lista estarão disponíveis no momento de execução da consulta.

4.3.1 Experimentos e Discussão dos Resultados

A fim de validarmos a SimSPARQL, definimos o cenário de testes dividido em dois tipos de experimentos, os quais serão apresentados abaixo.

Experimento 1

O propósito deste experimento consiste em avaliar quão confiável é a abordagem proposta, ou seja, verifica se as fontes de dados selecionadas são relevantes a cada consulta e se a estratégia de realizar o cálculo de similaridade é, de fato, eficaz. Os resultados obtidos com este experimento são apresentados na Tabela 4.1.

Logo, os campos da tabela apresentam informações referentes à inserção de cada consulta do experimento. A segunda coluna, nomeada como *Valor de Similaridade de cada consulta com cada grupo,* apresenta o valor de similaridade obtido através da relação entre os termos da consulta e os termos de cada grupo. A terceira coluna, nomeada como *Grupo com maior similaridade,* apresenta o grupo que forneceu maior valor de similaridade e pelo qual a consulta foi associada. A quarta coluna, chamada de *Dataset,* apresenta as fontes de dados capazes de responder a consulta ao mesmo tem poque está relacionada ao grupo que obtive maior similaridade. A penúltima coluna, denominada *Tipo da Consulta,* apresenta a abordagem utilizada pelo sistema para a

seleção das fontes, ou seja, o ASK representa o envio de consultas ASK a todas as fontes e o SimSPARQL informa que foi realizado o cálculo de similaridade. Por fim, a última coluna, nomeada como *Tempo para Seleção de Fontes*, apresenta o tempo gasto na seleção de fontes de dados relevantes a cada consulta.

Observe que no momento em que o tipo de consulta é ASK, um novo grupo é criado no sistema, podendo ser visualizado na coluna *Grupo com maior similaridade*. Isto acontece quando o valor de similaridade entre a consulta e os grupos existentes é menor do que o valor de *threshold*[16] configurado no sistema, não sendo possível afirmar que existe algum grupo similar que possua fontes de dados relevantes consulta. A coluna *Tempo* apresenta o tempo gasto na seleção de fontes de dados relevantes a cada consulta.

Tabela 4.1: Informações sobre os resultados da abordagem

Consulta	Valor de Similaridade de cada consulta com cada grupo	Grupo com maior similaridade	Datasets	Tipo de Consulta	Tempo para Seleção de Fontes
Consulta 01	Nenhum valor anterior	G1	2 e 27	ASK	40248 milisegundos
Consulta 02	G1 = 0,333	G2	2, 11, 13, 17 e 30	ASK	22081 milisegundos
Consulta 03	G1 = 0,250 G2 = 0,500	G3	5	ASK	22000 milisegundos
Consulta 04	G1 = 0,250 G2 = 0,500 G3 = 0,750	G3	5	SimSPARQL	2453 milisegundos
Consulta 05	G1 = 0,667 G2 = 0,333 G3 = 0,333	G1	2 e 27	SimSPARQL	1966 milisegundos
Consulta 06	G1 = 0,250 G2 = 0,500 G3 = 0,750	G3	5	SimSPARQL	2523 milisegundos
Consulta 07	G1 = 0 G2 = 0 G3 = 0	G4	3	ASK	21723 milisegundos
Consulta 08	G1 = 0,333 G2 = 0,667 G3 = 1 G4 = 0	G3	5	SimSPARQL	1155 milisegundos
Consulta 09	G1 = 0 G2 = 0 G3 = 0 G4 = 0,250	G5	3	ASK	22091 milisegundos
Consulta 10	G1 = 0 G2 = 0 G3 = 0 G4 = 0,250 G5 = 0,750	G5	3	SimSPARQL	1886 milisegundos
Consulta 11	G1 = 0,5 G2 = 1 G3 = 0,5	G2	2, 11, 13, 17 e 30	SimSPARQL	1138 milisegundos

[16]Threshold: configuramos o valor de 0,6 como threshold do sistema

	G4 = 0 G5 = 0				
Consulta 12	G1 = 0,250 G2 = 0,500 G3 = 0,750 G4 = 0 G5 = 0	G6	2	ASK	22586 milisegundos
Consulta 13	G1 = 0,250 G2 = 0,250 G3 = 0,500 G4 = 0 G5 = 0 G6 = 0,5	G7	2, 5, 8, 9, 11, 13, 17, 32	ASK	89182 milisegundos
Consulta 14	G1 = 0 G2 = 0 G3 = 1 G4 = 0 G5 = 0 G6 = 0,5 G7 = 0,5	G3	5	SimSPARQL	1113 milisegundos
Consulta 15	G1 = 0,250 G2 = 0,250 G3 = 0,500 G4 = 0 G5 = 0 G6 = 0,5 G7 = 0,5	G8	2	ASK	20332 milisegundos
Consulta 16	G1 = 0,250 G2 = 0,250 G3 = 0,500 G4 = 0 G5 = 0 G6 = 0,5 G7 = 0,5	G9	2	ASK	29269 milisegundos
Consulta 17	G1 = 0,400 G2 = 0,400 G3 = 0,600 G4 = 0 G5 = 0 G6 = 0,6 G7 = 0,4 G8= 0.4 G9 = 0.6	G10	2	ASK	22794 milisegundos
Consulta 18	G1 = 0,333 G2 = 0,333 G3 = 0,333 G4 = 0 G5 = 0 G6 = 0,333 G7 = 0 G8= 0 G9 = 0,667 G10 = 0.333	G9	2	SimSPARQL	2041 milisegundos
Consulta 19	G1 = 0,2 G2 = 0,4 G3 = 0,6 G4 = 0 G5 = 0 G6 = 0,6 G7 = 0,4 G8= 0,4 G9 = 0,667 G10 = 0.60	G9	2	SimSPARQL	1104 milisegundos
Consulta 20	G1 = 0,8 G2 = 0,2 G3 = 0,2 G4 = 0	G1	2 e 27	SimSPARQL	1772 milisegundos

	G5 = 0 G6 = 0,2 G7 = 0,2 G8= 0,4 G9 = 0,667 G10 = 0.2				
Consulta 21	G1 = 0,500 G2 = 0,250 G3 = 0,250 G4 = 0 G5 = 0 G6 = 0,250 G7 = 0,500 G8= 0 G9 = 0,500 G10 = 0.250	G11	2, 5 e 8	ASK	36397 milisegundos
Consulta 22	G1 = 0,500 G2 = 0,250 G3 = 0,250 G4 = 0 G5 = 0 G6 = 0,250 G7 = 0,500 G8= 0 G9 = 0,500 G10 = 0.250	G12	2, 13 e 32	ASK	79547 milisegundos
Consulta 23	G1 = 0 G2 = 0 G3 = 0 G4 = 0,250 G5 = 0,750 G6 = 0 G7 = 0 G8= 0 G9 = 0 G10 = 0 G11 = 0	G5	3	SimSPARQL	1743 milisegundos
Consulta 24	G1 = 0 G2 = 0 G3 = 0 G4 = 0,2 G5 = 0,2 G6 = 0 G7 = 0 G8= 0 G9 = 0 G10 = 0 G11 = 0 G12 = 0	G13	3	ASK	19592 milisegundos
Consulta 25	G1 = 0 G2 = 0 G3 = 0 G4 = 0,2 G5 = 0,6 G6 = 0 G7 = 0 G8= 0 G9 = 0 G10 = 0 G11 = 0 G12 = 0 G13 = 0,2	G5	3	SimSPARQL	3269 milisegundos
Consulta 26	G1 = 0 G2 = 0 G3 = 0 G4 = 0,667	G4	3	SimSPARQL	2167 milisegundos

	G5 = 0,333 G6 = 0 G7 = 0 G8= 0 G9 = 0 G10 = 0 G11 = 0 G12 = 0 G13 = 0,333				
Consulta 27	G1 = 0 G2 = 0 G3 = 0 G4 = 0,333 G5 = 0,667 G6 = 0 G7 = 0 G8= 0 G9 = 0 G10 = 0 G11 = 0 G12 = 0 G13 = 0,333	G5	3	SimSPARQL	1833 milisegundos
Consulta 28	G1 = 0,80 G2 = 0,2 G3 = 0 G4 = 0 G5 = 0 G6 = 0,2 G7 = 0,2 G8= 0,2 G9 = 0,2 G10 = o,2 G11 = 0,4 G12 = 0,6 G13 = 0	G1	2 e 27	SimSPARQL	1797 milisegundos
Consulta 29	G1 = 0,80 G2 = 0,2 G3 = 0 G4 = 0 G5 = 0 G6 = 0,2 G7 = 0,2 G8= 0,2 G9 = 0,2 G10 = o,2 G11 = 0,4 G12 = 0,6 G13 = 0	G3	5	SimSPARQL	2880 milisegundos
Consulta 30	G1 = 0,5 G2 = 0,333 G3 = 0,5 G4 = 0 G5 = 0 G6 = 0,5 G7 = 0,333 G8= 0,5 G9 = 0,167 G10 = o,500 G11 = 0,167 G12 = 0,667 G13 = 0	G12	2, 13 e 32	SimSPARQL	5685 milisegundos

Após a execução dos testes, foi possível visualizar como resultado alguns pontos:

(i) para todas as consultas testadas, a estratégia de similaridade forneceu resultados esperados, comprovando que o sistema se comportou adequadamente na extração e agrupamento dos termos das consultas, bem como na criação e atualização dos grupos de termos.

(ii) As fontes de dados relacionadas aos grupos de termos representativos das consultas possuíam informações relevantes às consultas agrupadas.

(iii) A medida em que novas consultas foram inseridas no sistema, houve uma estabilização na criação dos grupos de termos. Contudo, a lista de termos representativos de cada grupo estará sempre em atualização, dado que se uma nova consulta possuir um novo termo e as fontes do grupo forem capazes de respondê-lo, o termo será inserido no grupo em questão.

(iv) Verificou-se através da execução do experimento que a variação no tempo da seleção de fontes de dados pela SimSPARQL é extremamente menor se comparada à abordagem referente ao envio de consultas ASK.

(v) Há casos onde as consultas possuem termos similares, o valor de similaridade de algum dos grupos é maior do que o limiar, mas este novo termo não pode ser respondido pelas bases deste grupo, então, um novo grupo é criado para armazenar esta consulta. Assim, verificamos que no início da execução da abordagem, o número de grupos criados será crescente, sendo estabilizado a medida que novas consultas forem inseridas no sistema. Além disto, a execução da abordagem pode levar a grupos muitos similares até que seja atingido uma estabilização no sistema.

Experimento 2

O objetivo deste experimento é medir o tempo gasto na seleção de fontes de dados utilizando tanto a abordagem baseada em similaridade, proposta em nosso trabalho, quanto a abordagem baseada no envio de consultas ASK a todas as fontes de dados do sistema, proposta em FedX [Andreas *et al.,* 2011]. Para este experimento, consideramos o mesmo cenário de testes, constituído pelo conjunto de 30 consultas SPARQL e conjunto de fontes de dados. A Tabela 4.2 apresenta os valores de tempo gasto pelas duas abordagens, confirmando assim a eficácia da SimSPARQL.

Tabela 4.2: Tempo utilizado na seleção de Fontes de Dados

Consulta	**Abordagem baseada em Similaridade (SimSPARQL)**	**Abordagem baseada no envio de consultas ASK**
Consulta 01	----------	40248 milisegundos
Consulta 02	----------	22081 milisegundos
Consulta 03	----------	22000 milisegundos
Consulta 04	2453 milisegundos	29005 milisegundos
Consulta 05	1966 milisegundos	18050 milisegundos
Consulta 06	2523 milisegundos	17579 milisegundos
Consulta 07	----------	21723 milisegundos
Consulta 08	1155 milisegundos	18242 milisegundos
Consulta 09	----------	22091 milisegundos
Consulta 10	1886 milisegundos	18814 milisegundos
Consulta 11	1138 milisegundos	23964 milisegundos
Consulta 12	----------	22586 milisegundos
Consulta 13	----------	89182 milisegundos
Consulta 14	1113 milisegundos	17715 milisegundos
Consulta 15	----------	20332 milisegundos
Consulta 16	----------	29269 milisegundos
Consulta 17	----------	22794 milisegundos
Consulta 18	2041 milisegundos	18186 milisegundos
Consulta 19	1104 milisegundos	17306 milisegundos
Consulta 20	1772 milisegundos	110975 milisegundos
Consulta 21	----------	36397 milisegundos
Consulta 22	----------	79547 milisegundos
Consulta 23	1743 milisegundos	17630 milisegundos
Consulta 24	----------	19592 milisegundos
Consulta 25	3269 milisegundos	17893 milisegundos
Consulta 26	2167 milisegundos	18233 milisegundos
Consulta 27	1833 milisegundos	17584 milisegundos
Consulta 28	1797 milisegundos	18563 milisegundos
Consulta 29	2880 milisegundos	18509 milisegundos
Consulta 30	5685 milisegundos	17554 milisegundos

A partir dos resultados obtidos neste experimento, podemos concluir que, de modo geral, a seleção de fontes de dados promovida pela SimSPARQL foi executada em um tempo menor que a abordagem comparada. Isto significa que o tempo de resposta referente as duas abordagens são significativamente distintos e portanto, é viável a utilização da SimSPARQL na resolução do problema de seleção de fontes de dados em ambientes de federação de dados interligados.

Porém, temos que considerar que os resultados referentes as fontes de dados selecionadas podem ser distintos, ou seja, se tivermos duas consultas similares, onde

uma consulta é subconsulta de outra consulta mais geral, existe a possibilidade do resultado ser diferentes em função do grau de generalidade da consulta inserida primeiro no sistema. Assim, não podemos afirmar que todas as fontes de dados relacionadas à consulta geral estarão associadas ao grupo de termos referente à consulta geral. Todavia, isto pode ser considerado uma limitação da SimSPARQL mas é importante deixar claro que fontes de dados relevantes à consulta serão associadas à mesma.

Acreditamos que a economia de tempo de resposta na execução das consultas acontece devido ao fato de que calculando a similaridade entre os termos das consultas existentes nos grupos é menos custoso do que sempre consultar todas fontes de dados da federação. Na Figura 4.4 é possível visualizar o gráfico contendo as consultas utilizadas e comparadas neste segundo experimento, sendo exposta a comparação de tempo de resposta referentes às duas abordagens.

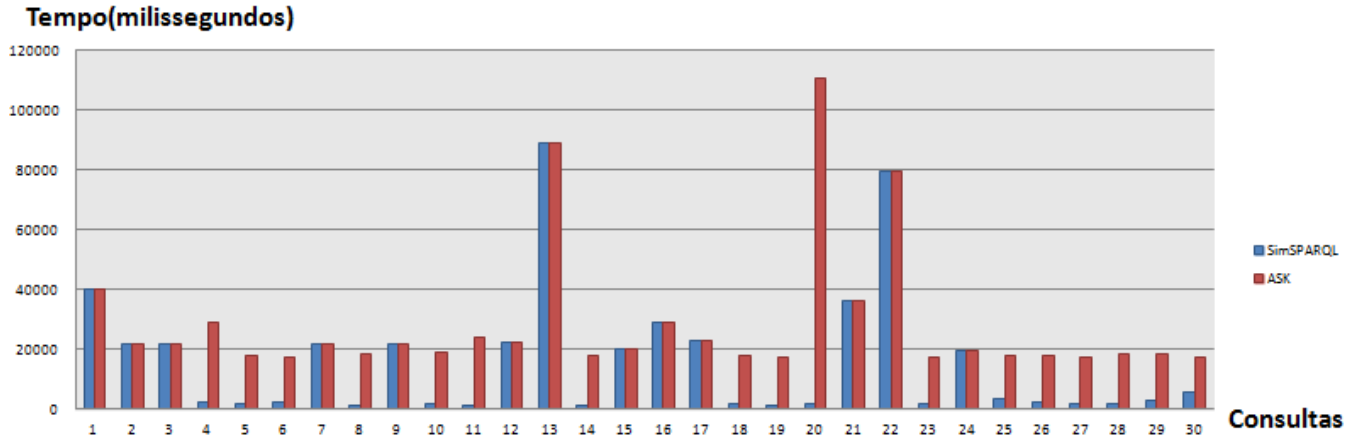


Figura 4.4: Relação de Tempo entre as estratégias SimSPARQL e ASK

Pode-se observar nesta mesma Figura que há casos onde o tempo gasto pelas duas abordagens são idênticos. Isto acontece porque quando não há consultas similares no sistema ou quando o valor de similaridade calculado é inferior ao configurado, a abordagem SimSPARQL se comporta semelhante ao envio de consultas ASK, consumindo o mesmo tempo na seleção das fontes de dados.

4.3.2 Algumas Considerações e Conclusões sobre os Experimentos

A partir dos resultados obtidos, podemos concluir que a SimSPARQL atende os requisitos ao qual se propõe, visto que comprovou-se a possibilidade de selecionar fontes de dados relevantes a uma dada consulta informada pelo usuário. No que diz respeito ao tempo de seleção de fontes, podemos observar que a SimSPARQL obteve

melhores resultados, uma vez que o valor do tempo de seleção de fontes para as consultas que possuíram grau de similaridade aceitável foi cerca de oito vezes menor do que o tempo de seleção gasto pela abordagem comparada.

Na Figura 4.5 é possível visualizar os resultados de tempo referente a execução das consultas no sistema. Os valores de tempo mais altos são referentes às consultas executadas através do envio de consultas ASK a todas as fontes de dados enquanto que os menores valores de tempo são referentes às consultas executadas seguindo a SimSPARQL.

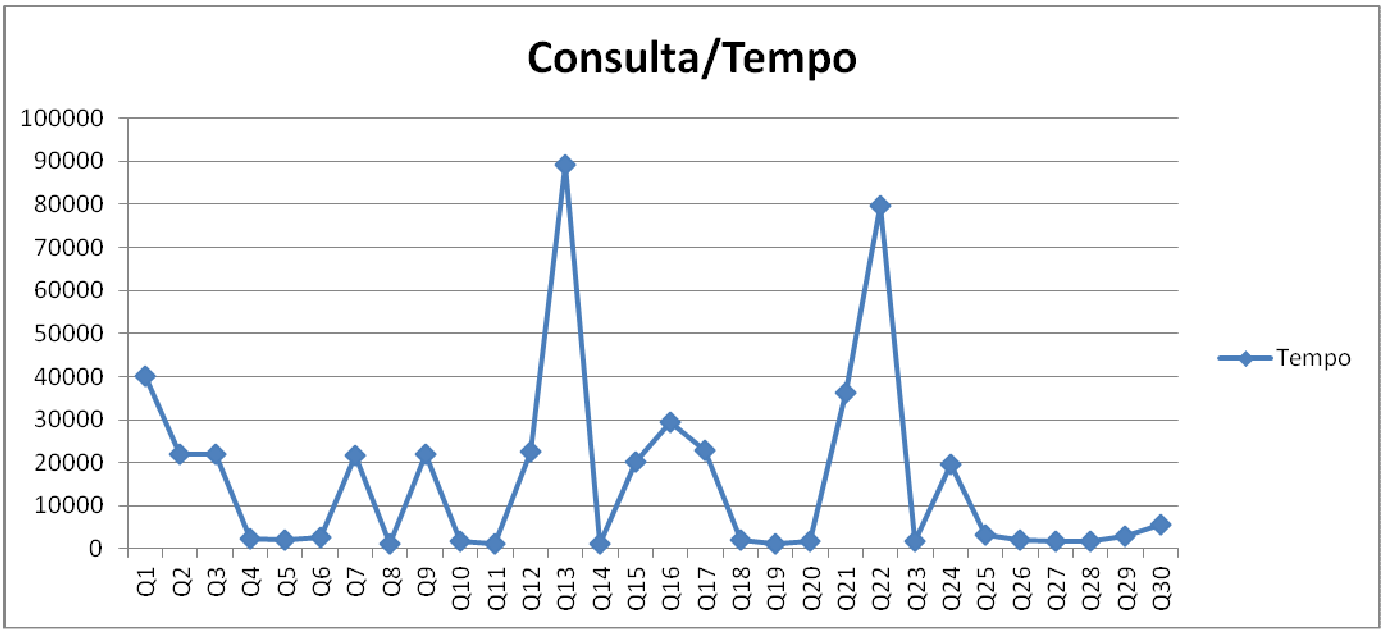


Figura 4.5: Tempo de Execução das consultas no sistema

A seguir, algumas considerações finais sobre os experimentos realizados:

(i) No Experimento 1, mostramos os resultados de algumas consultas submetidas ao sistema, a fim de verificar se a seleção de fontes de dados em um ambiente de federação de dados poderia ser realizada através da estratégia de similaridade entre os termos da consulta. Nestes resultados, demonstramos os valores de similaridade entre a consulta submetida e os grupos de fontes existente no sistema, apresentando qual grupo a consulta foi associada e quais as fontes de dados do grupo que serão capazes de responder tal consulta. O nosso objetivo foi mostrar que utilizar o método por meio de consultas ASK em conjunto com a verificação de similaridade dos termos da consulta é significativamente relevante na seleção de fontes de dados e menos custosa. Contudo, verificamos que no momento em que um novo grupo é criado e as

fontes de dados relevantes são associadas ao mesmo, somente o conjunto de termos pode crescer, o que torna a atualização das fontes de dados dos grupos uma limitação da nossa abordagem.

(ii) No Experimento 2, mostramos a comparação entre o tempo de execução gasto nas duas abordagens. Reforçamos que a SimSPARQL pode ser aplicada em um cenário onde não obtemos nenhuma informação sobre as fontes de dados. Então, verificamos por meio do Experimento 2 que a SimSPARQL realiza a seleção de fontes de dados de maneira eficiente, considerando como prioridade a diminuição de tempo e de esforço. Porém, a precisão ainda pode ser aperfeiçoada, visto que observamos que alguns grupos possuíam termos e fontes de dados semelhantes, mas que não se tornaram um único grupo em função da quantidade de termos extras presentes na consulta. Isto acontece quando temos consultas semelhantes semanticamente mas o valor de similaridade calculado entre os termos foi baixo, e por isso foram criados grupos de termos semanticamente semelhantes, ou seja, estes grupos possuíam parte dos termos iguais e as fontes de dados seriam as mesmas. Contudo, este problema será amenizado à medida que o número de consultas e termos são inseridos no sistema, uma vez que a quantidade de grupos formados tende a se tornar estável. A Figura 4.6 apresenta a estabilização na criação de grupos de termos no sistema, decorrentes da inserção de novas consultas no sistema.

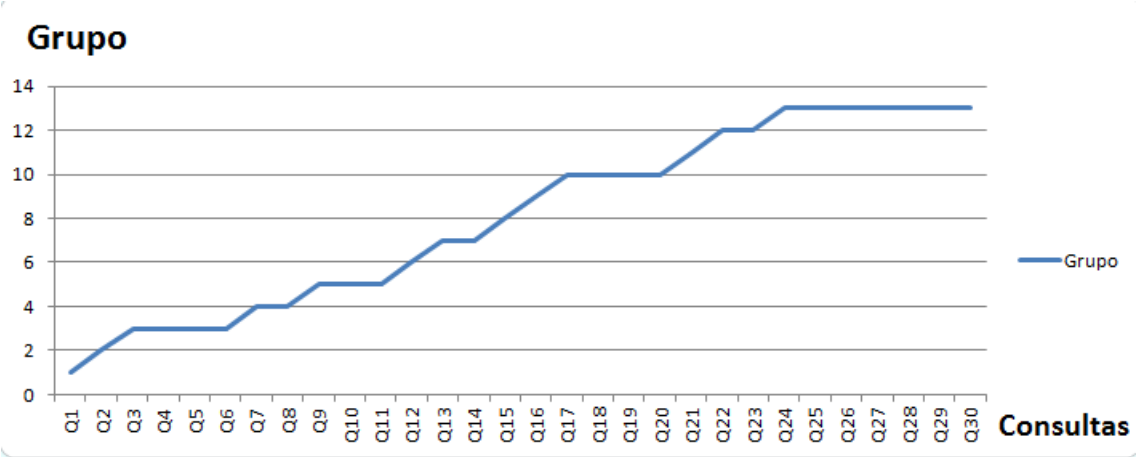


Figura 4.6: Estabilização na Criação de novos grupos

4.4 Considerações Finais

Neste capítulo apresentamos os aspectos de implementação e validação relacionados a ao nosso trabalho. A princípio, descrevemos a ferramenta SimSPARQL, a qual oferece funcionalidades para proporcionar ao usuário seleção de fontes de dados relevantes a uma dada consulta. Posteriormente, descrevemos pontos utilizados para validação da abordagem, expondo o cenário de testes ao qual os experimentos foram submetidos.

Assim, a execução dos experimentos teve como objetivo verificar se a abordagem se comportava de acordo com a sua proposta, apresentando como resultado os dados utilizados para seleção de fontes de dados em federações de dados interligados. Com isso, foram listadas informações referentes a análise do comportamento da SimSPARQL, tais como criação de grupos, valores de similaridade, conjunto de fontes de dados relevantes, a estratégia utilizada e o tempo de seleção de fontes obtido para cada consulta. Portanto, concluímos que para o conjunto de consultas testadas nos experimentos, a SimSPARQL obedece as suas funcionalidades e cumpre o objetivo de selecionar fontes de dados, além de comprovar um gasto significativamento menor de tempo no processo de seleção de fontes de dados.

Capítulo

5

Conclusões

O último capítulo desta dissertação apresenta as considerações finais relacionadas ao nosso trabalho, discutindo brevemente os principais pontos abordados ao longo de toda pesquisa. São apresentadas as contribuições e um resumo dos resultados obtidos, além de abordar as principais dificuldades e limitações. Por fim, oferece tópicos para aprofundamento e trabalhos futuros.

5.1 Considerações Finais

Esta dissertação apresentou a abordagem SimSPARQL, uma abordagem destinada a solucionar o problema de seleção de fontes de dados em ambientes de Federação de Dados interligados. A SimSPARQL adota uma estratégia baseada no cálculo de similaridade entre consultas SPARQL, de modo que são formados grupos de fontes de dados e grupos de termos representativos extraídos das consultas. Para validação de nossa abordagem, desenvolvemos um sistema capaz de comprovar a eficácia da SimSPARQL e analisamos as fontes de dados selecionadas para uma dada consulta. tendo em vista que o objetivo de selecionar fontes de dados relevantes a uma dada consulta, priorizando por um menor valor de tempo gasto para a seleção, foi atendido.

Assim, algumas considerações foram observadas na concepção, desenvolvimento e validação da SimSPARQL, as quais serão apresentadas a seguir:

- Definimos como estratégia a identificação de similaridade entre consultas SPARQL, obtida a partir da comparação entre os termos presentes nos padrões de

triplas de cada consulta inserida no sistema. Estes termos constituem os grupos de termos representativos diretamente relacionados a um conjunto de fontes de dados.

- Definimos que, para este trabalho, a SimSPARQL apenas tratará com o BGP da consulta, desconsiderando BGPs que possuem expressões de filtro.
- Para a realização dos experimentos desta dissertação, utilizamos a ontologia de domínio de dados bibliográficos fornecida pela *AKT Ontology*. As consultas SPARQL utilizadas nos experimentos desta dissertação foram construídas tendo como base os termos presentes nesta ontologia. Porém, nossa abordagem poderá ser aplicada em outras ontologias, necessitando de um conhecimento sobre os termos e conceitos definidos pelo vocabulário da mesma.
- Para manter uma maior confiabilidade do sistema, restringimos nossa abordagem no que diz respeito à atualização das fontes de dados de cada grupo de termos, onde só será possível associar fontes de dados a um grupo de termos no momento de sua criação. Ou seja, na prática, quando uma nova consulta é submetida ao sistema e esta for similar a um grupo existente, apenas o grupo de termos poderá crescer, e as fontes de dados associadas serão mantidas intactas.
- Para validação da SimSPARQL, montamos dois experimentos como cenários de testes, possibilitando a análise de informações como: valor de similaridade entre uma consulta e todos os grupos existentes, grupo de termos que será associado a consulta, conjunto de fontes capaz de respondê-la, tipo de estratégia utilizada para cada consulta submetida à ferramenta e o tempo gasto na seleção das fontes de dados. Para este estudo de caso, utilizamos como requisitos um conjunto de 30 consultas SPARQL(similares e não similares) e um conjunto de 33 *datasets* que utilizam o vocabulário da ontologia da AKT.

Por fim, verificamos a partir da análise dos resultados dos experimentos que a SimSPARQL se comportou conforme esperamos, uma vez que selecionou fontes de dados relevantes às consultas submetidas ao sistema, obtendo um menor tempo na seleção das mesmas.

5.2 Trabalhos Futuros

No desenvolvimento e validação da SimSPARQL, observamos alguns tópicos que possibilitarão o estudo e a execução de alguns trabalhos futuros, de modo que proporcione melhorias nos resultados obtidos, incluindo:

- Aperfeiçoar a abordagem para que considere BGPs com operações de filtros e modificadores como DISTINCT, ORDER BY e LIMIT, e operadores como AND, UNION, OPT, GRAPH e FILTER.
- Como a SimSPARQL só verifica a similaridade entre *strings,* relacionadas aos termos presentes nos padrões de triplas, apontamos como trabalho futuro a utilização de serviços para busca de URIs correspondentes, utilizando o *owl:sameas.*
- Realizar experimentos que façam uso de consultas que utilizem diversas ontologias de diferentes domínios, verificando o conjunto de fontes de dados selecionados e comparar o tempo de resposta utilizado pela SimSPARQL com outras abordagens existentes.
- Baseando-se nos tópicos de testes analisados nesta dissertação, um tópico futuro é ampliar o número de consultas, o número de ontologias utilizadas e o conjunto de datasets, a fim de analisar se a estratégia de similaridade proposta pela SimSPARQL se comporta da mesma maneira que a verificada neste trabalho.

REFERÊNCIAS

Arenas, M., Gutierrez, C., and P´erez, J. (2009). *On the semantics of sparql.* In Roberto De Virgilio, F. G. and Tanca, L., editors, Semantic Web Information Management: A Model Based Perspective, chapter 13. Spring, 1st edition.

Baeza-Yates, R.; Hurtado, C., And Mendoza M.. *Query Clustering For Boosting Web Page Ranking*. Advances In Web Intelligence, AWIC 2004, Springer LNCS, 3034, (2004) 164–17.

Beckett, D. and Berners-Lee, T.. *Turtle - terse rdf triple language.* Disponível em: <http://www.w3.org/TeamSubmission/turtle/>, 2008.

Beeferman, D. And Berger, A.. *Agglomerative Clustering Of A Search Engine Query Log*. In: KDD (1999) Boston, MA USA

Berners-Lee, T.; Hendler, J.; Lassila, O.. *The Semantic Web: A New Form Of Web Content That Is Meaningful To Computers Will Unleash A Evolution Of New Possibilities.* In: Scientific American Magazine. 2001. Disponível Em: <Http://Www.Cs.Nyes.Edu/Rgrimm/Teaching/Reading/Semantic-Web.Pdf>.

Biemann, C.. *Chinese Whispers - An Efficient Graph Clustering Algorithm And Its Application To Natural Language Processing Problems.* In: Association For Computational Linguistics, Pages: 73-80. 2006.

Bizer, C., Tom Health. *Linked Data – Evolving The Web Into A Global Data Space.* 2011.

Bizer, C. *The Web Of Linked Data: Architecture And Applications.* Freie Universitat Berlin. Webinale 2010.

Bizer, C.; Health, T.; Berners-Lee, T.. *The Story So Far*. In Heath, T., Hepp, M., And Bizer, C. (Eds.). Special Issue On *Linked Data*, International Journal On Semantic Web

And Information Systems (IJSWIS), 2009. Disponível Em: <Http://Linkeddata.Org/Docs/Ijswis-Special-Issue>

Cunha, D. R. B., Souza, D., Loscio, B. F. . *Linked Data*: Da Web De Documentos Para A Web De Dados. V Escola Regional Ceará, Maranhão E Piauí, 2011.

Haunsenblas, M.. *Exploiting Linked Data To Build Web Applications.* In: Journal IEEE Interne Computing Volume 13 Issue 4, July 2009.

Haase P., Mathab T., Ziller M.. *An Evaluation of Approaches to Federated Query Processing over Linked Data*. Proceeding of the 6th International Conference on Semantic Systems. 2010.

Harth, A.; Hose, K.; Karnstedt, M.; Polleres, A.; Sattle, K.; Umbrich, J.. *Data Summaries For On-Demand Queries Over Linked Data.* In: Proceedings Of The 19th International Conference On World Wide Web , 2010.

Hose K., Schenkel R.. *Towards Benefit-Based RDF Source Selection for SPARQL Queries*. In: 4th International Workshop on Semantic Web Information Management (SWIM 2012).

Hose K., Karnstedt M., Koch A., Sattler K., Zinn D.. *Processing Rank-Aware Queries in P2P Systems.* In: DBISP2P'05, p .238–249, 2005.

Hose K., Klan D. , Sattler K.. *Distributed Data Summaries for Approximate Query Processing in PDMS*. In: IDEAS '06, p .37–44, 2006

Huang, H., C. Liu, and X. Zhou. *Computing relaxed answers on RDF databases.* In: WISE, volume 5175 of LectureNotes in Computer Science. Springer, 2008. ISBN978-3-540-85480-7.

King, A. D.. *Graph Clustering With Restricted Neighbourhood Search.* Thesis For Degree Of Master Of Science In Department Of Computer Science. University Toronto, 2004.

Klyne, G. and J. J. Carroll (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax.* In: W3C Recommendation.

Kossmann, D.. *The state of the art in distributed query processing.* ACM Comput. Surv. 32(4) (2000) 422{469}.

Ladwig, G.; Tran, T.. Linked Data Query Processing Strategies - Technical Report. In: International Semantic Web Conference (1), volume 6496 of Lecture Notes in Computer Science. Springer, 2010. ISBN 978-3-642-17745-3

Langegger, W., Andreas. *SemWIQ - semantic web integrator and query engine.* In: GIJahrestagung (2), volume 134 of LNI. GI, 2008. ISBN 978-3-88579-228-4. URL http: //dblp.uni-trier.de/db/conf/gi/gi2008-2.html#LangeggerW07.

Morsey, M., Lehmann J., Auer S., Ngomo N. A.. *Dbpedia SPARQL Benchmark – Peerformance Assessment With Real Queries On Real Data.* In: ISWC. 2011.

Ngomo, N. A-C.; Schumacher, F.. *Borderflow: A Local Graph Clustering Algorithm For Natural Language Processing.* In: Computational Linguistics And Intelligent Text Processing, Pages 547–558, 2009.

Olaf Hartig, Christian Bizer, and Johann-Christoph Freytag. *Executing SPARQL Queries over the Web of Linked Data.* In: ISWC 2009. Springer, 2009.

Pérez, j., Arenas M., Gutierrez C.. *Semantics and Complexity of SPARQL.* Universidad de Talca - Chile. Universidad de Chile. 2009.

Prud'hommeaux, E.; Seaborne, A.. *SPARQL Query Language For RDF*. In: W3C Recommendation, 2008.

Prud'hommeaux, E.. *Optimal RDF access to relational databases.* Disponível em: <http://www.w3.org/2004/04/30-RDF-RDB-access/>, April 2004

Quilitz, B., U. Leser. *Querying distributed RDF data sources with SPARQL.* In: Proceedings of the 5th European Semantic Web Conference, LNCS. Springer Verlag, Berlin,Heidelberg, 2008.

Raimond, Y. *Linked Data On The BBC*. In: BBC Future Media & Tecnology For Audio And Music And Mobile. Disponível Em: Http://Www.Slideshare.Net/Moustaki/Linked-Data-On-The-Bbc-2638734

Reddy, K. B. R., Kumar, P. S. *Efficient approximate SPARQL querying of Web of Linked Data.* Indian Institute of Tecnology Madras, Chennai, India. 2012.

Sahami, M., Heilman, T.D.. *A Web-Based Kernel Function For Measuring The Similarity Of Short Text Snippets.* In: World Wide Web Conference (2006) 377–386.

Sakr, S.; Al-Naymat, G.. *Efficient Relational Techiques For Processing Graph Queries.* In: Journal Of Computer Science And Technology. Vol. 25, Num. 6, 1237-1255. 2010.

Scaniello, G.; Marcus, A. *Clustering Support For Static Concept Location In Source Code.* In: 19th IEEE International Conference On Program Comprehension. 2011.

Schwarte A., Haase P., Hose K., Schenkel R., Schmidt M.. *FedX: Optimization Techniques for Federated Query Processing on Linked Data.* In: ESWC Poster and Demo Session Proceedings. Springer, 2011.

Volz, J.; Bizer, C.; Gaedke, M.; Kobilarov, G.. *Discovering And Maintaining Links On The Web Of Data.* In: The 8th International Semantic Web Conference (ISWC), 2009.

Vilar, B. S. C. M. Processamento De Consultas Baseado Em Ontologias Para Sistemas De Biodiversidade. Campinas, 2009. 67f Dissertação (Mestrado Em Ciências Da Computação) Instituto De Computação. Universidade Estadual De Campinas – UNICAMP.

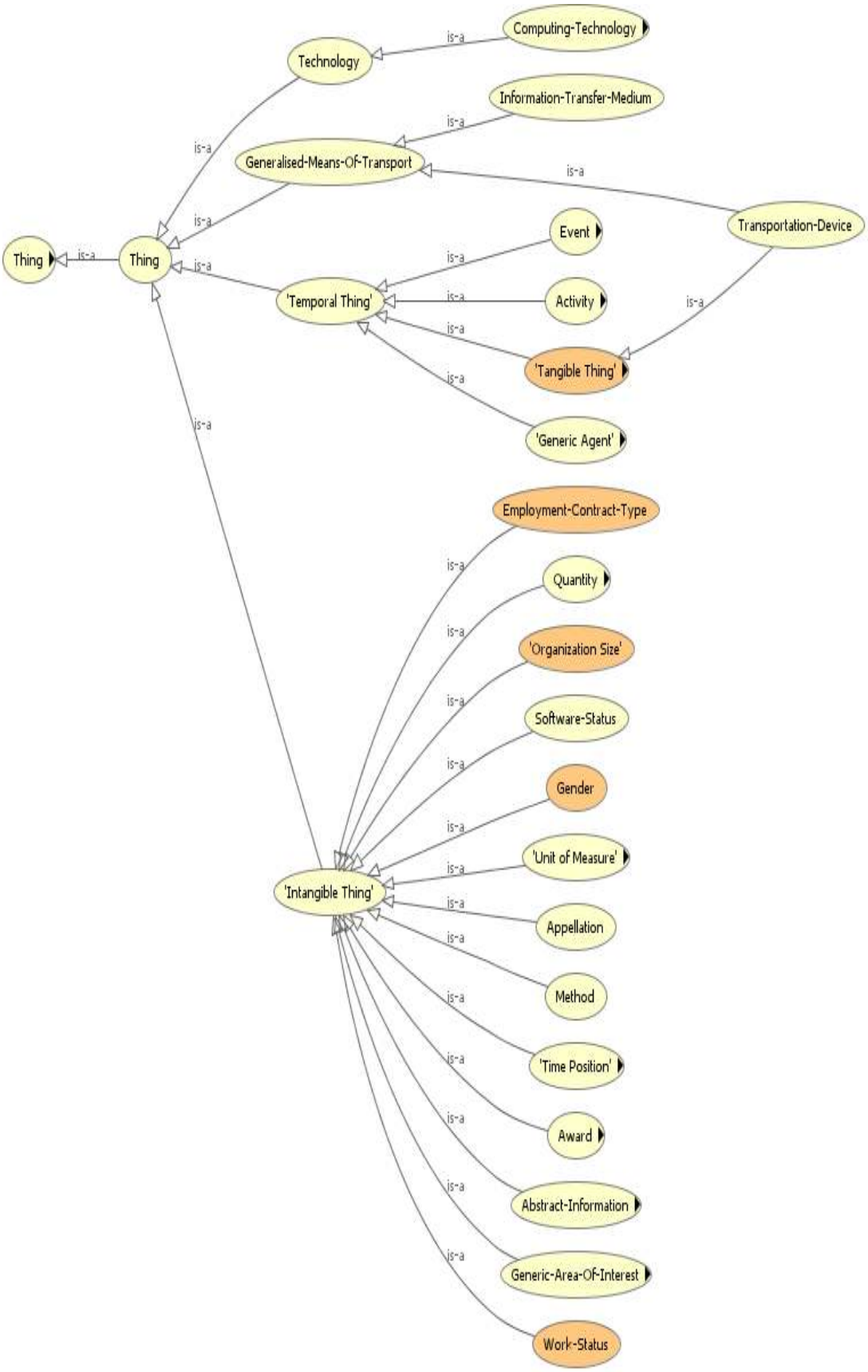
Wang, H., T. Penin, K. Xu, J. Chen, X. Sun, L. Fu, Q. Liu, Y. Yu, T. Tran, P. Haase, and R. Studer, *Hermes: a travel through semantics on the data web.* In: SIGMOD Conference.2009.

W3C. *Semantic Web Activity.* (2001). Disponível Em: <Http://Www.W3.Org/2001/12/Semwebfin/W3cs>.

Watson, M.. *Pratical Semantic Web And Linked Data Applications*. 2011.

Wen, J.; Mie, J.; Zhang, H.. *Clustering User Queries Of A Search Engine*. In: Proc. At 10th International World Wide Web Conference, W3C (2001)

ANEXO A – Ontologia AKT



ANEXO B - Conjunto de *Datasets*

ID	*Dataset*	*Endpoint*	*Quantidade de triplas*
1	Open Archive Initiative Harvest over OAI-PMH (RKBExplorer)	http://oai.rkbexplorer.com/sparql/	24206591
2	DBLP Computer Science Bibliography (RKBExplorer)	http://dblp.rkbexplorer.com/sparql/	24112294
3	dotAC (RKBExplorer)	http://dotac.rkbexplorer.com/sparql/	21279220
4	Korean Institute of Science Technology and Information (RKBExplorer)	http://kisti.rkbexplorer.com/sparql/	12815726
5	Association for Computing Machinery (ACM) (RKBExplorer)	http://acm.rkbexplorer.com/sparql/	12402336
6	National Science Foundation (RKBExplorer)	http://nsf.rkbexplorer.com/sparql/	11822283
7	ePrints3 Institutional Archive Collection (RKBExplorer)	http://eprints.rkbexplorer.com/sparql/	8417840
8	CiteSeer (Research Index) (RKBExplorer)	http://citeseer.rkbexplorer.com/sparql/	8146852
9	Research Assessment Exercise 2001 (RKBExplorer)	http://rae2001.rkbexplorer.com/sparql/	2718105
10	UN/LOCODE (RKBExplorer)	http://unlocode.rkbexplorer.com/sparql/	371549
11	RISKS Digest (RKBExplorer)	http://risks.rkbexplorer.com/sparql/	322913
12	School of Electronics and Computer Science, University of Southampton (RKBExplorer)	http://southampton.rkbexplorer.com/sparql/	298067
13	Université Paul Sabatier - Toulouse 3 (RKB Explorer)	http://irit.rkbexplorer.com/sparql/	176542
14	Ordnance Survey (RKBExplorer)	http://os.rkbexplorer.com/sparql/	161227
15	ERA - Australian Research Council publication ratings (RKBExplorer)	http://era.rkbexplorer.com/sparql/	157376
16	IEEE Papers (RKBExplorer)	http://ieee.rkbexplorer.com/sparql/	91564
17	University of Newcastle upon Tyne (RKBExplorer)	http://newcastle.rkbexplorer.com/sparql/	87505
18	UK JISC (RKBExplorer)	http://jisc.rkbexplorer.com/sparql/	64672
19	ReSIST Resilience Mechanisms (RKBExplorer.com)	http://resex.rkbexplorer.com/sparql/	55582
20	LAAS-CNRS (RKBExplorer)	http://laas.rkbexplorer.com/sparql/	52312
21	Resilient Computing Courseware (RKBExplorer)	http://courseware.rkbexplorer.com/sparql/	45780
22	IBM Research GmbH (RKBExplorer)	http://ibm.rkbexplorer.com/sparql/	44721
23	ReSIST Project Wiki	http://wiki.rkbexplorer.com/sparql/	44379

	(RKBExplorer)		
24	Università di Pisa (RKBExplorer)	http://pisa.rkbexplorer.com/sparql/	43219
25	Budapest University of Technology and Economics (RKBExplorer)	http://budapest.rkbexplorer.com/sparql/	42378
26	Università degli studi di Roma "La Sapienza" (RKBExplorer)	http://roma.rkbexplorer.com/sparql/	41313
27	DEPLOY (RKBExplorer)	http://deploy.rkbexplorer.com/sparql/	41071
28	Institut Eurécom (RKBExplorer)	http://eurecom.rkbexplorer.com/sparql/	40863
29	Universität Ulm (RKBExplorer)	http://ulm.rkbexplorer.com/sparql/	40044
30	France Telecom Recherche et Développement (RKBExplorer)	http://ft.rkbexplorer.com/sparql/	39843
31	Deep Blue (RKBExplorer)	http://deepblue.rkbexplorer.com/sparql/	39421
32	ReSIST MSc in Resilient Computing Curriculum (RKBExplorer)	http://curriculum.rkbexplorer.com/sparql/	37789
33	Technische Universität Darmstadt (RKBExplorer)	http://darmstadt.rkbexplorer.com/sparql/	35325

APENDICE A - Consultas Utilizadas no Experimento

Consulta 01

Linguagem Natural

Retorna os títulos de todos os artigos de *Journal* que aconteceram no ano de 2011.

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?articleTitle ?journalTitle
WHERE {
 ?journal rdf:type akt:Journal .
 ?journal akt:has-title ?journalTitle .
 ?article akt:article-of-journal ?journal .
 ?article akt:has-title ?articleTitle .
 ?article akt:has-date akt-date:2011 .
 }
```

Consulta 02

Linguagem Natural

Retorna o título, o autor e o volume de todas as *Magazines.*

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>

SELECT ?nomeMagazine ?volume ?nomeAutor
WHERE{
  ?magazine akt:has-title  ?nomeMagazine .
  ?magazine akt:has-author  ?nomeAutor .
  ?magazine akt:has-volume ?volume.
}
```

Consulta 03

Linguagem Natural

Retorna os artigos que tiveram como autora *Bernadette Farias Lóscio.*

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>

SELECT ?nomeArtigo
WHERE {
?artigo akt:has-title ?nomeArtigo .
?artigo akt:has-author ?nomeAutor .
?nomeAutor akt:full-name "Bernadette Farias Loscio" .
}
```

Consulta 04

Linguagem Natural

Retorna os artigos que tiveram como autor *Patrick C. Fischer.*

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>

SELECT ?nomeArtigo ?x ?artigo
WHERE {
?artigo akt:has-title ?nomeArtigo .
?artigo akt:has-author ?nomeAutor .
?nomeAutor akt:full-name “Patrick C. Fischer” .
 }
```

Consulta 05

Linguagem Natural

Retorne o título de todas as publicações do ano de 2010.

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?publicationTitle
WHERE{
?publication   akt:has-title   ?publicationTitle .
?publication   akt:has-date   akt-date:2010 .
}
```

Consulta 06

Linguagem Natural

Retorna o título dos artigos publicados pelo autor *Alon Halevy.*

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?articleTitle
WHERE {
  ?article akt:has-author ?halevy .
  ?halevy akt:full-name "Alon Y. Halevy" .
  ?article akt:has-title ?articleTitle .
}
```

Consulta 07

Linguagem Natural

Retorna o nome e o telefone da instituição que possui como referência a URI *<http://dotac.rkbexplorer.com/id/org-anglia-ac-uk>*

Linguagem SPARQL

```
PREFIX id:  <http://dotac.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?telefone
WHERE
{
 id:org-anglia-ac-uk   rdfs:label    ?nomeInstituicao .
 id:org-anglia-ac-uk   foaf:phone  ?telefone .
}
```

Consulta 08

Linguagem Natural

Retorna os dados referentes à todas as publicações científicas, tais como nomes dos artigos, o nome do autor e a uri do artigo.

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeArtigo ?nomeAutor ?artigo
WHERE {
?artigo akt:has-title ?nomeArtigo .
?artigo akt:has-author ?autor .
?autor akt:full-name ?nomeAutor .
}
```

Consulta 09

Linguagem Natural

Retorna o nome, a cidade e o código postal da Universidade que possui como referência a URI *<http://dotac.rkbexplorer.com/id/org-bournemouth-ac-uk>*.

Linguagem SPARQL

```
PREFIX id:  < http://dotac.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?cidade ?codigoPostal
WHERE
{
 id:org-bournemouth-ac-uk rdfs:label ?nomeInstituicao.
 id:org-bournemouth-ac-uk foaf:city ?cidade.
 id:org-bournemouth-ac-uk foaf:postcode ?codigoPostal.
}
```

Consulta 10

Linguagem Natural

Retorna o nome, a cidade e o código postal da Universidade que possui como referência a URI <*http://dotac.rkbexplorer.com/id/org-icr-ac-uk*>.

Linguagem SPARQL

```
PREFIX id:  < http://dotac.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?cidade ?codigoPostal
WHERE
{
 id:org-icr-ac-uk rdfs:label ?nomeInstituicao.
 id:org-icr-ac-uk foaf:city ?cidade.
 id:org-icr-ac-uk foaf:postcode ?codigoPostal.
}
```

Consulta 11

Linguagem Natural

Retorna o título e o volume de todas as revistas.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT  ?nomeMagazine ?volume
WHERE {
?magazine akt:has-title  ?nomeMagazine .
?nomeMagazine akt:has-volume ?volume.
}
```

Consulta 12

Linguagem Natural

Retorne o nome do autor cuja publicação é "*On formalisms for Turing machines*".

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeAutor
WHERE {
?artigo akt:has-title "On formalisms for Turing machines" .
?artigo akt:has-author ?autor .
?autor akt:full-name ?nomeAutor .
}
```

Consulta 13

Linguagem Natural

Retorna os nomes de todas as pessoas que estão como autores de no minimo dois artigos.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT  ?name
WHERE {
 ?article1 rdf:type akt:Article-Reference .
 ?article2 rdf:type akt:Article-Reference .
 ?article1 akt:has-author ?person .
 ?article2 akt:has-author ?person .
 ?person akt:full-name ?name .
}
```

Consulta 14

Linguagem Natural

Retornar todos as informações relacionadas a *Alon Halevy*.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?subject ?predicate
WHERE {
 ?subject ?predicate ?halevy .
 ?halevy akt:full-name "Alon Y. Halevy" .
}
```

Consulta 15

Linguagem Natural

Retorna o nome dos autores e o ano da publicação do livro "*A Semantic web primer*", referente a URI < *http://dblp.rkbexplorer.com/id/books/daglib/0011076*>.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT DISTINCT ?authorName ?year WHERE {
 <http://dblp.rkbexplorer.com/id/books/daglib/0011076> akt:has-author ?author .
 ?author akt:full-name ?authorName .
 <http://dblp.rkbexplorer.com/id/books/daglib/0011076> akt:has-date ?year .
}
```

Consulta 16

Linguagem Natural

Retorna os artigos que citam como referencia a publicação de Anais da Conferência "*ACM SIGMOD International Conference on Management of Data*", SIGMOD, Atenas, Grécia, Junho 12-16, 2011.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?articleTitle WHERE {
 ?article akt:cites-publication-reference <http://dblp.rkbexplorer.com/id/conf/sigmod/2011> .
 ?article akt:has-title ?articleTitle .
}
```

Consulta 17

Linguagem Natural

Retorne os artigos que foram publicados no "*Journal of Research and Practice in Information Technology"*.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?articleTitle ?authorName
WHERE {
 ?article akt:article-of-journal <http://dblp.rkbexplorer.com/id/journals-
a7141f673c4edbe5841eeeba7e0d4a21> .
 ?article akt:has-title ?articleTitle .
 ?article akt:has-author ?author .
 ?author akt:full-name ?authorName .
}
```

Consulta 18

Linguagem Natural

Retorna os artigos que citam como referencia a publicação
7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, 7-9 Setembro, 2011.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?articleTitle
WHERE {
 ?article akt:cites-publication-reference <http://dblp.rkbexplorer.com/id/conf/i-semantics/2011> .
 ?article akt:has-title ?articleTitle .
}
```

Consulta 19

Linguagem Natural

Retorna o nome dos autores que publicaram no 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, 7-9 Setembro, 2011.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?authorName
WHERE {
 ?article akt:cites-publication-reference <http://dblp.rkbexplorer.com/id/conf/i-semantics/2011> .
 ?article akt:has-title ?articleTitle .
 ?article akt:has-author ?author .
 ?author akt:full-name ?authorName .
}
```

Consulta 20

Linguagem Natural

Retorne todos os *Proceedings* dos eventos que ocorreram em 2011.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?conferenceProceedingTitle
WHERE {
  ?conferenceProceeding rdf:type akt:Conference-Proceedings-Reference.
  ?conferenceProceeding akt:has-date akt-date:2011 .
  ?conferenceProceeding akt:has-title ?conferenceProceedingTitle .
}
```

Consulta 21

Linguagem Natural

Retorne os títulos dos artigos em que um autor cita outro autor em suas publicações.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?articleTitle1 ?articleTitle2
WHERE {
  ?article1 rdf:type akt:Article-Reference .
  ?article2 rdf:type akt:Article-Reference .
  ?article1 akt:cites-publication-reference ?article2.
  ?article1 akt:has-title ?articleTitle1 .
  ?article2 akt:has-title ?articleTitle2 .
}
```

Consulta 22

Linguagem Natural

Retorne o título dos artigos e os autores que publicaram artigos no ano de 2007.

Linguagem SPARQL

```
PREFIX id:  < http://dblp.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT  ?authorName ?authorTitle
WHERE {
  ?article rdf:type akt:Article-Reference .
  ?article akt:has-date akt-date:2007 .
?article1 akt:has-title ?articleTitle .
  ?article akt:has-author ?author .
  ?author akt:full-name ?authorName .
}
```

Consulta 23

Linguagem Natural

Retorne o nome, a cidade e o código postal da universidade que possui como referência a URI < *http://dotac.rkbexplorer.com/id/org-bbk-ac-uk*>.

Linguagem SPARQL

```
PREFIX id:  < http://dotac.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?cidade ?codigoPostal
WHERE
{
 id:org-bbk-ac-uk rdfs:label ?nomeInstituicao.
 id:org-bbk-ac-uk foaf:city ?cidade.
 id:org-bbk-ac-uk foaf:postcode ?codigoPostal.
}
```

Consulta 24

Linguagem Natural

Retorna o nome, a latitude e a longitude da universidade que possui como referência a URI <*http://dotac.rkbexplorer.com/id/org-jisc-ac-uk*>.

Linguagem SPARQL

```
PREFIX id:      <http://dotac.rkbexplorer.com/id/>
PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:     <http://www.w3.org/2002/07/owl#>
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX dc:      <http://purl.org/dc/elements/1.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX geo:  <http://www.w3.org/2003/01/geo/wgs84_pos#>

SELECT ?nomeUniversidade ?latitude ?longitude
WHERE
{id:org-jisc-ac-uk rdfs:label ?nomeUniversidade.
 id:location-bs81qu geo:lat ?latitude.
 id:location-bs81qu geo:long ?longitude. }
```

Consulta 25

Linguagem Natural

Retorna a cidade e o codigo postal da universidade nomeada por *University of Bedfordshire.*

Linguagem SPARQL

```
PREFIX id:  < http://dotac.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?cidade ?codigoPostal
WHERE
{
 ?nomeInstituicao rdfs:label "University of Bedfordshire" .
 id:org-beds-ac-uk foaf:city ?cidade.
 id:org-beds-ac-uk foaf:postcode ?codigoPostal.
}
```

Consulta 26

Linguagem Natural

Retorna o telefone da intituição de nome *University College Birmingham.*

Linguagem SPARQL

```
PREFIX id:  < http://dotac.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?telefone
WHERE
{
 ?nomeInst rdfs:label “University College Birmingham” .
 ?nomeInst foaf:phone ?telefone.
}
```

Consulta 27

Linguagem Natural

Retorna o nome da Instituição e da cidade que possui como referência a URI < *http://dotac.rkbexplorer.com/id/org-bcu-ac-uk*>.

Linguagem SPARQL

```
PREFIX id:  < http://dotac.rkbexplorer.com/id>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeInstituicao ?cidade ?codigoPostal
WHERE
{
 id:org-bcu-ac-uk rdfs:label ?nomeInstituicao.
 id:org-bcu-ac-uk foaf:city ?cidade.
}
```

Consulta 28

Linguagem Natural

Retorne todos os *Proceedings* dos eventos que ocorreram no ano de 2008.

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?conferenceProceedingTitle
WHERE {
  ?conferenceProceeding rdf:type akt:Conference-Proceedings-Reference.
  ?conferenceProceeding akt:has-date akt-date:2008 .
  ?conferenceProceeding akt:has-title ?conferenceProceedingTitle .
}
```

Consulta 29

Linguagem Natural

Retorna o nome do Artigo publicado pela autora *Susan L. Epstein*.

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT  ?authorName ?articleTitle
WHERE {
 ?article akt:cites-publication-reference ?nameArticle .
 ?nameArticle akt:has-title ?articleTitle .
 ?article akt:has-author ?author .
 ?author akt:full-name "Susan L. Epstein" .
}
```

Consulta 30

Linguagem Natural

Retorna o nome dos artigos publicados por Seymour Ginsburg nos Proceedings dos eventos que aconteceram em 2011.

Linguagem SPARQL

```
PREFIX id:  <http://dblp.rkbexplorer.com/id/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX akt:  <http://www.aktors.org/ontology/portal#>
PREFIX owl:  <http://www.w3.org/2002/07/owl#>
PREFIX akt-date: <http://www.aktors.org/ontology/date#>

SELECT ?nomeArtigo
WHERE {
?artigo akt:has-title ?nomeArtigo.
?artigo akt:has-author ?autor .
?autor akt:full-name “Seymour Ginsburg”.
?conferenceProceeding akt:has-date akt-date:2011 .
}
```