SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL E TECNOLOGIA INDUSTRIAL

Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação de mestrado

Modelo computacional para analisar dados semiestruturados na Web Semântica com o auxílio da teoria de redes

Apresentada por: Gabriela Oliveira Mota da Silva
Orientador: Prof. Dr. Hernane Borges de Barros Pereira
Co-orientador: Prof. Dr. Eduardo Manuel de Freitas Jorge

Outubro de 2014

Gabriela Oliveira Mota da Silva

Modelo computacional para analisar dados semiestruturados na Web Semântica com o auxílio da teoria de redes

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Interdisciplinar

Orientador: Prof. Dr. Hernane Borges de Barros Pereira
SENAI CIMATEC

Co-orientador: Prof. Dr. Eduardo Manuel de Freitas Jorge
SENAI CIMATEC

Salvador
SENAI CIMATEC
2014

Nota sobre o estilo do PPGMCTI

Esta dissertação de mestrado foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e estão disponíveis em formato eletrônico (*download* na Página Web http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html ou solicitação via e-mail à secretaria do programa) e em formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do programa de pós-graduação supracitado.

SENAI CIMATEC

Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial

Mestrado em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, leram e recomendam a aprovação [com distinção] da Dissertação de mestrado, intitulada "Modelo computacional para analisar dados semiestruturados na Web Semântica com o auxílio da teoria de redes", apresentada no dia 17 de outubro de 2014, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Orientador:

Prof. Dr. Hernane Borges de Barros Pereira
SENAI CIMATEC

Co-orientador:

Prof. Dr. Eduardo Manuel de Freitas Jorge
SENAI CIMATEC

Membro externo da Banca:

Prof. Dr. Ed Porto Bezerra
Universidade Federal da Paraíba

Membro interno da Banca:

Prof. Dr. Roberto Luiz Souza Monteiro
SENAI CIMATEC

Dedico este trabalho a minha família, alicerce da vida.

Agradecimentos

Agradeço aos professores Hernane Pereira e Eduardo Jorge, pela rica orientação e paciência.

Aos professores do PPG MCTI, pela passagem valiosa de conhecimentos ao longo da jornada acadêmica.

Aos colegas da quarta turma de Mestrado Acadêmico do SENAI CIMATEC e do grupo de pesquisa Fuxicos e Boatos, dos quais me distanciei, porém sinto muitas saudades.

À Shankar pela parceria no desenvolvimento da pesquisa.

Ao noivo, companheiro e cúmplice Cleber, pelo apoio na difícil decisão de me mudar para Brasília e, mesmo assim, não desistir de me formar.

À família, amigos e todos aqueles que contribuíram direta ou indiretamente para este trabalho, meus sinceros agradecimentos.

Salvador, Brasil — Gabriela Oliveira Mota da Silva
17 de Outubro de 2014

Resumo

A ciência das redes é uma área de pesquisa em ascensão. Seus métodos e ferramentas permitem que pesquisadores de diversas áreas analisem dados a partir das propriedades das conexões existentes entre eles. Por outro lado, a Web Semântica disponibiliza uma gama de bases de dados semiestruturados, em linguagem aberta na Web. Este trabalho tem como objetivo propor um modelo computacional para extração e transformação de dados semiestruturados na Web Semântica para um formato que permita a interoperabilidade com ferramentas de redes complexas, e.g. Gephi. Assim, por meio deste modelo, pesquisadores poderão minerar bases de dados abertos da Web Semântica e efetuar suas análises sob a perspectiva da teoria de redes. O presente trabalho segue uma metodologia de pesquisa quantitativa, devido à análise de efetividade ser baseada em índices de redes coletados. Espera-se que este trabalho facilite o acesso a dados robustos disponíveis na Web e, assim, contribua com os avanços de pesquisas em diversas áreas do conhecimento.

Palavras-chave: RDF, Web Semântica, Linked Data, Redes sociais e complexas, Interoperabilidade de dados

Abstract

The science of networks is a research area on the rise. His methods and tools allow researchers from various fields to analyze data from the properties of existing connections between them. Furthermore, the Semantic Web offers a set of open language semi-structured databases, on the Web. This paper aims to propose a computational model for extraction and processing of Semantic Web semi-structured data to a format that enables interoperability with Complex Networks tools, such as Gephi. Thus, through this model, researchers may be able to mine Semantic Web open databases and perform their analysis from the perspective of networks theory. This study follows a quantitative research methodology, due to the effectiveness analysis be based on collected networks properties. We expect that this work will smooth access to robust data available on the web and will contribute on the progress of researches in various fields of knowledge.

Keywords: Linked Data, World Wide Web, Data interchanging, Semantic Web, Social and complex networks

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Siglas

CIMATEC ..	Centro Integrado de Manufatura e Tecnologia
GEXF	Graph Exchange XML Format
IFBA	Instituto Federal da Bahia
JSON	JavaScript Object Notation
HTML	Hypertext MarkupLanguage
HTTP	Hypertext TransferProtocol
LOD	Linking Open Data
PDF	Portable Document Format
PPGMCTI ..	Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial
RDF	Resource Description Framework
REST	REpresentational State Tranfer
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
SENAI	Serviço Nacional de Aprendizagem Industrial
SOAP	Simple Object Access Protocol
SPARQL	SPARQL Protocol and RDF Query Language
SQL	Structured Query Language
UFBA	Universidade Federal da Bahia
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language

Capítulo Um

Introdução

> O maior inimigo do conhecimento não é a ignorância; é a ilusão do conhecimento.
>
> (Stephen Hawking)

Durante a última década, uma série de trabalhos deu início a uma ciência das redes, um novo campo de pesquisa, com seu próprio conjunto de desafios e realizações peculiares (BARABÁSI, 2009). Outra área que cresceu especialmente na última década diz respeito ao estudo de modelos, métodos e ferramentas para publicação de dados semiestruturados na Web, dando forma à Web Semântica. Por um lado, para se estudar as redes sociais e complexas, há um trabalho inicial de obtenção da informação, muitas vezes custoso. Um exemplo consiste na busca por informações cronológicas sobre a Segunda Guerra Mundial em sites da Web. Por outro lado, a Web Semântica disponibiliza uma gama de bases de dados semiestruturados, em linguagem aberta. Ou seja, é possível que haja arquivos publicados sobre o mesmo assunto (Segunda Guerra Mundial), porém acompanhados de metadados interpretáveis por programas de computador, não apenas por humanos. Tais programas podem fazer uso de algoritmos automatizados para ler esses documentos e converter os dados semânticos para formatos de apresentação que valorizem a informação almejada pelos pesquisadores.

Uma importante forma de apresentação é dada através de softwares de análise de redes, que facilitam a interpretação gráfica e matemática de propriedades que caracterizam as redes complexas. Porém, até o presente momento, não há protocolos ou ferramentas largamente difundidos para auxiliar nas tarefas de filtragem e conversão de dados semiestruturados para formatos que possam ser interpretados por tais softwares.

O presente projeto visa preencher a lacuna que separa os dois campos de pesquisa supracitados: Web Semântica e teoria de redes, com a criação de um modelo ou protocolo para análise de bases de dados semiestruturadas e sua conversão, de forma a prover aos pesquisadores meios de analisar os dados sob a luz da teoria das redes.

Esta pesquisa está inserida na grande área dos sistemas complexos e, mais especificamente, das redes complexas. Também está inserida no campo de estudo dos Sistemas de Representação do Conhecimento, uma vez que se propõe a analisar modelos de criação, organização, gestão e difusão do conhecimento.

1.1 Definição do problema

O campo de pesquisa de sistemas complexos pode ser considerado como uma recente área de estudo. Em paralelo, a ciência das redes emerge nesse contexto, alavancada pelos avanços tecnológicos, que hoje permitem a coleta e processamento de grandes massas de dados em tempo razoável. Por outro lado, a Web Semântica é uma proposta de evolução da World Wide Web, onde as páginas são acrescidas de metadados, i.e. dados sobre os dados. Entre outros benefícios, como interoperabilidade entre sistemas da Web e melhorias nos motores de busca, a Web Semântica provê uma base de dados semiestruturada que cobre diversas áreas do conhecimento. Essa base de dados pode ser caracterizada como um modelo de criação, organização, gestão e difusão do conhecimento, mais especificamente, como uma rede semântica.

Assim, partindo do princípio que a Web Semântica é também um conjunto de redes semânticas, surgem as seguintes questões de pesquisa: como essas redes semânticas podem ser apresentadas aos pesquisadores de forma a facilitar suas pesquisas? Que contribuição teria sua conversão para formatos de redes complexas? Como essas bases de dados da Web Semântica podem contribuir com pesquisas científicas e sociais, a partir dos estudos das características inerentes às redes complexas?

1.2 Objetivo

O objetivo deste trabalho consiste em propor um modelo computacional para extração e transformação de dados semiestruturados na Web Semântica para um formato que permita a interoperabilidade com ferramentas de redes complexas, tais como Gephi e Pajek. Espera-se alcançar tal objetivo através das seguintes metas ou objetivos específicos:

- Definir um modelo genérico capaz de coletar quaisquer tipos de dados estruturados, semiestruturados ou não estruturados e convertê-los para formatos de redes sociais e complexas;
- Restringir o modelo para utilizar como dados de entrada quaisquer ontologias presentes em bases de dados semiestruturados em Linked Data (e.g. DBpedia);
- Estabelecer um protocolo (i.e. conjunto de regras) baseado em RDF, para extrair os conceitos e suas relações - sujeitos, predicados e objetos - desses *datasets*, considerando a estrutura da rede em relação ao seu domínio de aplicação;
- Criar um método de conversão dos conceitos e relações extraídos em formatos - nós e arestas - que possam ser interpretados por ferramentas de análise de redes (e.g. Gephi, Pajek);

- Implementar uma instância do modelo - solução computacional - e definir um domínio de aplicação para verificação do modelo;
- Verificar a eficiência do modelo, através da caracterização das relações semânticas e análise dos parâmetros de uma rede complexa construída a partir do domínio escolhido;
- Propor melhorias no modelo, de forma a imprimir robustez ao sistema de representação do conhecimento;
- Propor outras linhas de análise e abordagem do problema.

1.3 Importância da pesquisa

Este trabalho possui relevância em diferentes campos de estudos científicos e sociais, uma vez que se propõe a auxiliar na coleta, processamento e análise de dados de naturezas diversas, espalhados pela Web de forma semiestruturada e, assim, prover subsídios para que pesquisadores possam interpretar os dados e utilizá-los em suas pesquisas. Este processo permite que os pesquisadores transportem seu domínio de estudo de um sistema de representação do conhecimento para outro, no caso, da ótica das ontologias para a teoria das redes complexas.

Outro aspecto relevante é a aplicação dos resultados deste trabalho na melhoria dos motores de busca e recuperação de dados e informação na Web.

1.4 Limites e limitações

Este trabalho não se propõe a construir uma ferramenta comercial, mas limita-se ao modelo computacional de arquitetura genérica, que permite implementações de acordo com o tipo de dado que se deseja garimpar e a ferramenta de rede onde se deseja analisar os dados. Entretanto, como forma de validar o trabalho, foi construída uma implementação da arquitetura, que utiliza consultas SPARQL para extrair dados de bases RDF e gera redes no formato de arquivo da ferramenta GEPHI.

1.5 Aspectos metodológicos

O presente trabalho segue uma metodologia de pesquisa quantitativa, pois a efetividade do modelo é validada por um experimento, onde alguns índices da rede gerada são calculados

e comparados com os índices obtidos por meio de outros modelos. Segundo Liebscher (1998), os métodos quantitativos são apropriados quando é possível quantificar as variáveis de interesse, onde as hipóteses podem ser formuladas e testadas, e inferências podem ser projetadas a partir de amostras para as populações. Esse viés se torna um facilitador no processo de validação em trabalhos cuja importância está na coleta e quantificação dos dados, para se chegar a resultados objetivos e com evidência empírica.

Em contrapartida, métodos qualitativos poderiam ser utilizados para realizar o mapeamento das relações semânticas entre os elementos de um sistema de criação, organização, gestão e difusão do conhecimento. Porém, neste trabalho, o contexto em que o modelo está inserido não possui relevância, assim, o uso da metodologia qualitativa não é adequado. Dito isto, convém detalhar as atividades empregadas por esta metodologia, para se chegar ao objetivo deste trabalho:

- Levantamento bibliográfico das categorias teóricas, que envolvem os assuntos: modelos de criação, organização, gestão e difusão do conhecimento; Web Semântica; RDF e Linked Data; redes semânticas; redes complexas; ferramentas para análise de redes;
- Definição de um modelo genérico, baseado em etapas ou módulos, onde os descritores de coleta e conversão de dados possam ser acrescentados, de forma independente;
- Estabelecimento de um protocolo ou método de coleta de dados baseado em RDF. O método visa extrair os conceitos e suas relações de um *dataset* semiestruturado, de forma parametrizada, utilizando consultas SPARQL;
- Estabelecimento de um protocolo ou método de conversão dos elementos formais catalogados para um formato que possa ser interpretado por ferramentas de análise de redes;
- Verificação do modelo proposto através de experimentos, que consistem na implementação de uma instância do modelo (ferramenta RDFree), definição de um domínio de estudo presente na Web Semântica, coleta dos dados e conversão para o formato de redes;
- Análise e caracterização topológica da rede a partir de cálculos dos índices mais relevantes para o domínio de estudo. Este estudo deve validar o modelo de criação, organização, gestão e difusão do conhecimento gerado a partir da ferramenta RDFree e permitir que se publique os resultados.

1.6 Organização da Dissertação de mestrado

Este documento apresenta 5 capítulos e está estruturado da seguinte forma:

- **Capítulo 1 - Introdução**: contextualiza o âmbito, no qual a pesquisa proposta está inserida. Apresenta, portanto, a definição do problema, objetivos e justificativas da pesquisa e como esta dissertação de mestrado está estruturada;
- **Capítulo 2 - Teoria de redes: uma síntese**: traz uma revisão bibliográfica sobre redes complexas, dissertando sobre as descobertas na área, de forma cronológica;
- **Capítulo 3 - Web Semântica**: apresenta a revisão bibliográfica sobre o outro tema desta dissertação: Web Semântica, também apresentando os fatos mais relevantes da área de forma cronológica. O capítulo apresenta, ainda, as tecnologias da Web Semântica relacionadas a este trabalho;
- **Capítulo 4 - Modelo proposto**: tem como objetivo descrever o modelo proposto, através de seus requisitos e arquitetura, bem como apresentar um experimento que comprove sua eficiência, dentro do cenário a que se propõe. Neste capítulo ainda são discutidos os resultados obtidos com o experimento;
- **Capítulo 5 - Considerações finais**: Apresenta as conclusões, contribuições e algumas sugestões de atividades de pesquisa a serem desenvolvidas no futuro.

Capítulo Dois

Teoria de redes: uma síntese

> A ciência nos ofereceu uma explicação de como a complexidade (o difícil) surgiu como resultado da simplicidade (o fácil).
>
> (Richard Dawkins)

Nos últimos anos, um campo de estudo multidisciplinar tem se desenvolvido rapidamente, com objetivo de estudar os sistemas interconectados encontrados na biologia, sociologia e ciência da computação, dentre outras áreas. A ciência das redes estuda as conexões entre elementos de um sistema, onde a complexidade de suas interações o torna maior do que apenas a soma de suas partes (NUSSENSVEIG, 2008). Muitas dessas redes possuem também grande escala, o que dificulta seu entendimento de forma visual, carecendo de modelos de propriedades matemáticas. Historicamente, a topologia das redes de larga escala tem sido explicada através do modelo de grafos aleatórios de Erdös e Rényi (1959). Após décadas de descobertas, foram propostos os modelos mundo-pequeno de Watts e Strogatz (1998) e livre de escala de Barabási e Albert (1999). Hoje muitos sistemas reais são estudados a partir das redes sociais e complexas, a exemplo dos trabalhos seguintes:

Um estudo que utiliza elementos de redes sociais para modelar a propagação de doenças infecciosas (WATTS; STROGATZ, 1998). Em uma rede de pessoas saudáveis, no tempo $t = 0$ é introduzido um indivpiduo infectado. A cada iteração de tempo, cada indivíduo infectado pode transmitir a doença para seu vizinho saudável com probabilidade r. Indivíduos infectados permanecem na rede por uma iteração de tempo e depois são removidos pemanentemente (por imunidade ou morte). os resultados indicam que as doenças infecciosas se propagam mais rapidamente em uma rede *small-world* (será abordada na subseção 2.2.2) do que em redes aleatórias (ver subseção 2.2.1). A Figura 2.1 compara a distribuição de probabilidades do modelo em relação à rede aleatória.

Outra importante publicação em redes sociais é o livro de Wasserman e Faust (1994), que enfatiza o uso de redes como modelos de estudos em ciências sociais e comportamentais. Nesta publicação, os autores introduzem o conceito de redes de dois modos, onde os eventos, ou conexões, podem estar ligados a sobgrupos de atores de tipos (ou modos) diferentes. Assim, é possível estudar as propriedades em redes heterogêneas e inferir os fenômenos que ocorrem com ênfase nas interações entre os subgrupos de atores em estudo. E.g. em uma rede com dois conjuntos de atores: corporações e organizações não governamentais, é possível estudar o fluxo de investimentos que parte das corporações para as organizações não governamentais. As redes de afiliações são um tipo especial de rede de

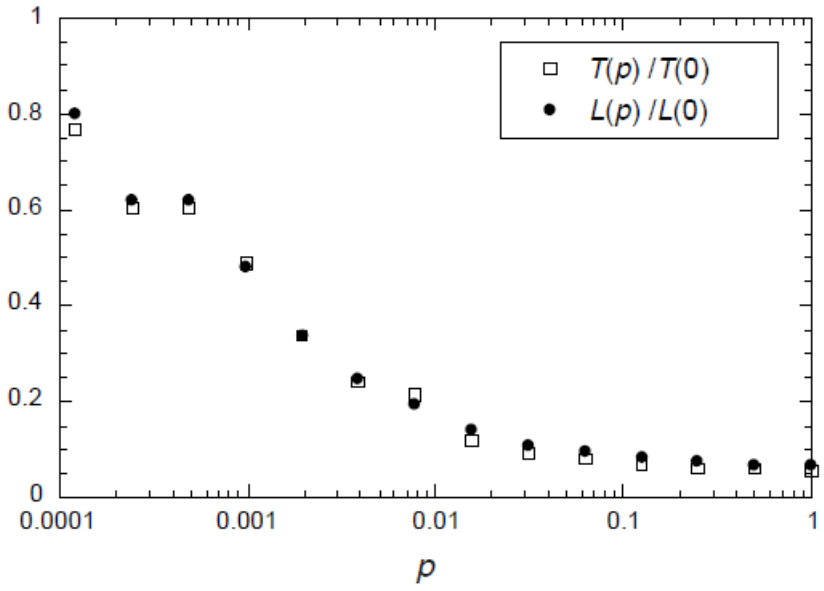

Figura 2.1: O tempo $T(P)$ necessário para que uma doença com probabilidade máxima de infecção ($r = 1$) se espalhe por toda a população tem essencialmente a mesma forma funcional que o comprimento do caminho característico $L(p)$. Mesmo que apenas uma pequena percentagem das arestas da estrutura original sejam religadas aleatoriamente, o tempo para a infecção global é quase tão curto quanto para um grafo aleatório. Fonte: (WATTS; STROGATZ, 1998).

dois modos onde o segundo modo é um conjunto de eventos aos quais os atores pertencem, e.g. eventos formais, como clubes, organizações voluntárias e conselhos de diretores de empresas, ou informais, como uma festa ou a simples obeservação das interaçãos entre um grupo de pessoas com o mesmo objetivo.

Uma série de outros trabalhos na área podem ser consultados para melhor entendimento da teoria de redes sociais e complexas. A exemplo da revisão de Newman (2003), que contém um relato crítico, ao discutir uma lista de referências no assunto e apresentar um quadro exaustivo de propriedades estruturais e modelos de redes, além de abordar o aspecto dinâmico das redes. Outra publicação completa no campo é o estudo de Boccaletti et al. (2006) que, além de revisar o assunto, aborda novas descobertas na área, como o conceito de redes ponderadas, que introduz valores ou pesos para medir a força das conexões da rede. Outro conceito introduzido é o de redes espaciais (*spatial networks*), que são modelos onde as conexões de longo alcance são limitadas pela distância euclidiana[1] ou a quantidade de conexões é limitada pelo espaço físico disponível para ligar os nós, e.g. redes planas de sistemas de ruas urbanas e redes aéreas limitadas pelo espaço dos aeroportos. O estudo mostra que esta característica inflige consequências importantes sobre as propriedades estatísticas da rede.

[1]Em matemática, distância euclidiana é a distância entre dois pontos, que pode ser provada pela aplicação repetida do teorema de Pitágoras.

Além da extensa lista de publicações científicas, há também publicações introdutórias que apresentam as descobertas na área de forma cronológica, voltadas para o público leigo, a exemplo dos livros Linked: A Nova Ciência dos Networks (BARABÁSI, 2009), Seis Graus de Separação: a evolução da ciência de redes em uma era conectada (WATTS, 2009) e Nexus: fundamentos da ciência dos networks (BUCHANAN, 2010), todos com edições publicadas em Português pela editora Leopardo entre os anos de 2009 e 2010. Este último apresenta a visão de um jornalista científico sobre o assunto.

O presente capítulo apresenta uma introdução aos principais conceitos que permeiam as redes sociais e complexas, e que se iniciam pela teoria dos grafos. Com a evolução da ciência das redes, os grafos deixaram de ser uma linguagem utilizada para descrever modelos abstratos e passaram a ser uma ferramenta para modelagem e análise de dados interconectados empíricos (NEWMAN; BARABÁSI; WATTS, 2006). E embora muitas propriedades aplicadas à análise de redes tenham evoluído da teoria dos grafos, segundo Newman (2003) os métodos estatísticos desenvolvidos recentemente mudaram a abordagem da comunidade científica, ao alavancar a aplicação de métricas em redes de larga escala.

2.1 Teoria dos grafos

A história da teoria dos grafos remonta ao ano de 1736, quando o matemático Leonhard Euler (1707-1783) interessou-se pelo enigma das pontes de Königsberg. Tratava-se de uma cidade cortada pelo rio Prególia, no território da antiga Prussia, onde sua geografia formava duas ilhas, ligadas às demais áreas de terra por sete pontes. O problema consistia em descobrir um caminho que cruzasse as sete pontes exatamente uma vez, i.e. sem passar pela mesma ponte mais de uma vez. A solução proposta por Euler consistiu em representar cada pedaço de terra separado pelas pontes como pontos ou nós e cada ponte por uma linha, ou aresta, ligando dois nós. A Figura 2.2 foi citada do trabalho original de Euler e esquematiza a geografia de Königsberg, com suas sete pontes e quatro pedaços de terra, representados por letras maiúsculas.

Euler, assim, construiu o primeiro grafo de que se tem história e resolveu o problema ao demonstrar que não existe tal caminho se o grafo possui mais de dois nós com número ímpar de arestas. A explicação é simples: para se conseguir uma passagem contínua que atravesse todas as pontes, deve haver um único ponto de partida e um de chegada, portanto, apenas dois nós podem ter número ímpar de pontes (BARABÁSI, 2009). No grafo do problema de Königsberg, havia quatro nós com número ímpar de arestas, conforme demonstra a Figura 2.3.

As ideias de Euler deram início à teoria dos grafos, que enuncia: um grafo $G(V, \mathcal{E})$ é um

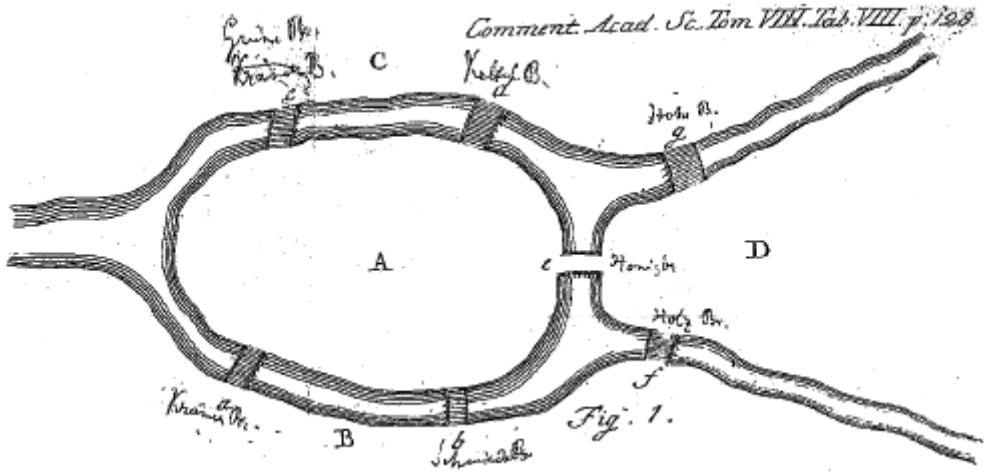

Figura 2.2: As Sete Pontes de Königsberg. Fonte: (EULER, 1736).

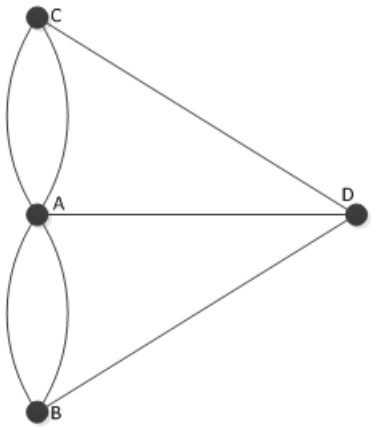

Figura 2.3: Grafo representativo do problema de Königsberg, onde cada pedaço de terra é um nó (letras maiúsculas) e cada ponte é uma aresta. Fonte: Elaborada pela autora.

conjunto finito de pontos $V(G)$ =$\{v_1,...,v_n\}$ - que são vértices - e um subconjunto $\mathcal{E}$ = $\{(e_1,..., e_m\}$ formado por pares desordenados de vértices de $V(G)$ - que são as arestas (BOLLOBAS, 1998). Uma aresta que liga os vértices v_1 e v_2 é representada por $e = v_1v_2$ e um grafo completamente conectado é aquele em que todos os seus vértices estão ligados por arestas. Assim, o grafo conectado $G\alpha(V, \mathcal{E})$, onde $V(G_\alpha) = \{v_1, v_2, v_3\}$ terá exatamente o conjunto de arestas $\mathcal{E}(G_\alpha) = \{(v_1, v_2), (v_1, v_3), (v_2, v_3)\}$. Grafos direcionados são formados por conjuntos ordenados de arestas e, por isso, possuem indicação dos nós de início e de fim para cada aresta. Isso permite que o mesmo par de nós possua diversas arestas paralelas. Já o grafo representativo das pontes de Königsberg (Figura 2.3) é chamado de multigrafo, pois, além de permitir múltiplas arestas entre seus nós, estas não possuem direcionamento. Ele pode ser representado de forma simplificada por $G_\beta(V, \mathcal{E})$, onde $V(G_\beta) = \{A, B, C, D\}$ e $\mathcal{E}(G_\beta) = \{(A, B), (A, C), (A, D), (B, D), (C, D)\}$.

A contribuição mais importante da prova de Euler para a teoria dos grafos é que a existência do caminho não depende de qualquer esforço para encontrá-lo, mas sim, trata-se de uma propriedade do grafo. Outras propriedades dos grafos foram estudadas e documentadas posteriormente, e.g.: i) ordem $|V|$ é o número de vértices do grafo (chamaremos daqui pra frente de $n(G)$ ou simplesmente n); ii) grau k de V é a quantidade de arestas adjacentes ao vértice V, i.e. que se ligam a ele. Por consequência, em grafos direcionados, é necessário fazer distinção entre o grau de entrada e o de saída de cada vértice, de acordo com o direcionamento das arestas (DIESTEL, 2010); iii) a partir da propriedade grau, é possível calcular o grau médio $\langle k \rangle$ do grafo, através da equação 2.1.

$$\langle k \rangle = \frac{2m}{n} \tag{2.1}$$

Graças às contribuições de Euler para a teoria dos grafos, nomeou-se Euleriano um grafo conectado que contém um caminho entre o vértice v_i e o vértice $v_j \neq v_i$ se v_i e v_j são os únicos vértices de grau ímpar. Euler também provou ser condição suficiente para a existência de um caminho que visite todos os nós apenas uma vez em um grafo com $|V|$ > 3 vértices, a condição de que todos os vértices tenham grau par (BOLLOBAS, 1998). Se, além disso, o caminho inicia e termina no mesmo vértice, este é chamado Ciclo Euleriano.

Na Königsberg atual (hoje Caliningrado, na Rússia), uma nova ponte foi construída entre B e C, o que incrementou a quantidade de arestas desses dois nós para quatro. Assim, apenas os nós A e D têm número ímpar de arestas, tornando o Ciclo Euleriano possível nessa nova configuração. E assim como a geografia de Königsberg foi modificada, também seriam as aplicações dos grafos. Mais de duzentos anos depois, o matemático Paul Erdös engendraria outros significados aos grafos de Euler, ao dar início a uma nova ciência: a ciência das redes.

2.2 Redes Complexas

O estudo de Paul Erdös e Alfred Rényi (ERDöS; RéNYI, 1959) é apontado na literatura como um dos precursores para o advento da ciência das redes. A partir de então, os grafos passaram a ser utilizados para modelar algo mais além de enigmas curiosos; passaram a modelar situações do mundo real de importância científica. Sua utilização em diversas áreas do conhecimento ganha importância à medida que favorece a descoberta de propriedades ocultas do domínio em estudo, antes invisíveis aos olhos dos pesquisadores.

Em teoria dos grafos, grafos regulares são aqueles que possuem o mesmo grau em todos os seus vértices. De forma análoga, diz-se que redes regulares possuem todos os vértices com mesmo grau. As redes complexas possuem propriedades emergentes que as diferem de grafos comuns. Ao longo desta seção, tais propriedades serão expostas, à medida que as descobertas no campo de pesquisa das redes complexas são apresentadas, em ordem cronológica.

2.2.1 Redes aleatórias

Os modelos de grafos aleatórios foram desenvolvidos independentemente por Solomonoff R.; Rapoport (1951) e Erdös e Rényi (1959) e consistem na topologia mais fundamental para modelagem de redes complexas. O modelo publicado em 1959 por Erdös e Rényi (1959) e complementado em 1960 (ERDöS; RéNYI, 1960) ficou conhecido como modelo Erdös-Rényi, ou ER. Segundo os autores, há dois modos de se construir um grafo aleatório. O primeiro modelo é representado por $G(n, m)$, onde n é o número de nós e m é uma quantidade fixa de conexões adicionadas de maneira aleatória. Significa que para gerar a rede aleatória, é preciso escolher um grafo $G(n, m)$ entre todas as $C_{(n,m)} = \binom{\binom{n}{2}}{m}$ combinações possíveis de grafos, com igual probabilidade de escolha (ERDöS; RéNYI, 1959).

O segundo modelo de Erdös e Rényi evidencia a evolução da rede baseada em probabilidade p única para todos os nós. Neste modelo, representado por $G(n, p)$, adiciona-se uma nova aresta com probabilidade p independente, aplicada às $\binom{n}{2} = \frac{n(n-1)}{2}$ conexões possíveis. Em redes de grande tamanho ($n \to \infty$) o número médio de conexões de cada vértice k é dado por $z = p(n-1)$ (ERDöS; RéNYI, 1960). Por exemplo, para construir uma rede com 100 vértices e probabilidade $p = 0{,}05$, seguindo o modelo $G(n, p)$, é necessário considerar cada par de nós e aplicar a probabilidade de conexão. O resultado é semelhante ao gráfico exibido na Figura 2.4, obtido após ajustar as cores e tamanhos dos nós de acordo com seu grau.

A partir do valor do grau de cada nó da rede (conforme visto na seção 2.1), é possível definir

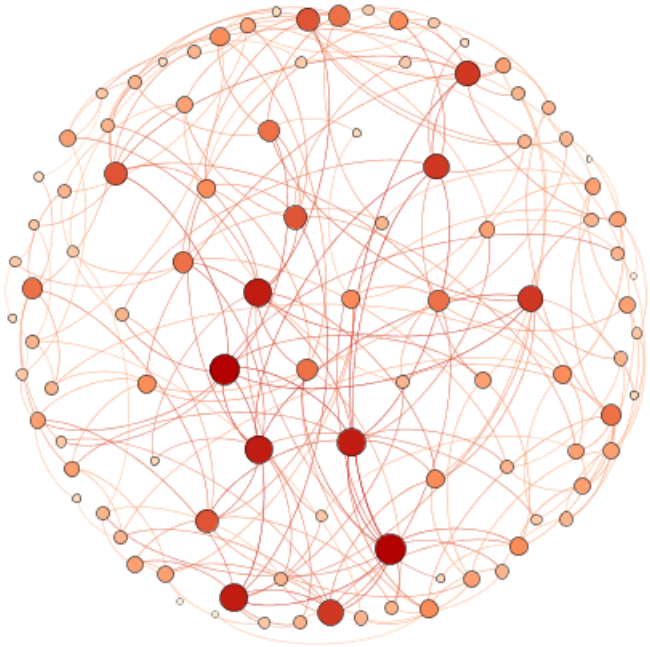

Figura 2.4: Exemplo de rede aleatória construída a partir do modelo $G(n, p)$, com n=100 e p=0,05. Fonte: Elaborada pela autora.

a propriedade da distribuição de graus, que é a função de distribuição da probabilidade de um dado vértice, escolhido ao acaso na rede, ter determinado grau. Para redes aleatórias, a distribuição de graus comporta-se como uma binomial, que se aproxima da distribuição de Poisson quando $n \to \infty$ (ERDöS; RéNYI, 1960; NEWMAN, 2003; ALBERT; BARABÁSI, 2002):

$$P_k = \binom{n}{k} p^k (1-p)^{n-k} \simeq \frac{z^k e^{-z}}{k!} \tag{2.2}$$

A Figura 2.5 apresenta o gráfico da distribuição de graus da rede supracitada como exemplo.

Ao utilizar a equação 2.1 para calcular o grau médio da rede, obtém-se o valor $\langle k \rangle = 2,42$, o que significa que os nós da rede têm em média 2,42 arestas ou conexões. Já a densidade Δ mede o "poder de relacionamento" entre os vértices da rede. Trata-se da razão entre o número de arestas existentes e o número de arestas possíveis na rede, ou seja, o número de vértices n tomados 2 a 2: $\binom{n}{2}$. Seus valores variam desde 0, quando não existem arestas no grafo, até 1, quando todas as arestas possíveis estão presentes. Ou seja, quanto mais arestas e rede tiver, mais densa ela será. A equação para calcular a densidade da rede é dada por:

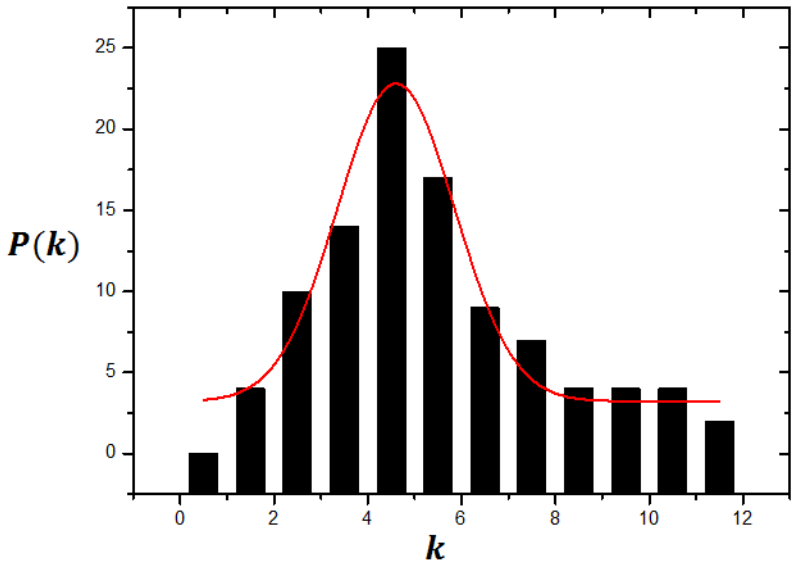

Figura 2.5: Gráfico da distribuição de graus de uma rede aleatória construída a partir do modelo $G(n, p)$, com n=100 e p=0,05. Fonte: Elaborada pela autora.

$$\Delta = \frac{m}{\frac{n(n-1)}{2}} \tag{2.3}$$

A rede da Figura 2.4 possui $\Delta = 0{,}049$, ou seja, é uma rede pouco densa. Isso se deve ao valor da probabilidade aplicada às conexões da rede aleatória ter sido baixo ($p = 0{,}05$).

Mas apenas o número de arestas que ligam dois nós vizinhos é suficiente para caracterizar uma rede? Ao analisar os vizinhos dos vizinhos e a forma como eles se conectam é possível inferir aspectos da topologia (estrutura) e do comportamento (dinâmica) da rede. De acordo com Newman (2003), a propriedade transitividade ou aglomeração (*clustering*, em Inglês) significa a probabilidade de que dois vértices que são vizinhos de outro vértice na rede sejam também vizinhos entre si. Na linguagem de redes sociais, significa que o amigo do seu amigo provavelmente é seu amigo também. O valor médio da aglomeração na rede é calculado com base no número de triângulos (três vértices conectados entre si) em relação ao total de caminhos de tamanho dois, confome indica a equação:

$$\langle C \rangle = \frac{6 \times \ \textit{número de triângulos na rede}}{\textit{número de caminhos de tamanho dois}} \tag{2.4}$$

onde caminho de tamanho dois se refere a um caminho direcionado, iniciando em um vértice específico e passando por outros dois vértices. Em redes aleatórias, esse coeficiente é baixo, chegando próximo ao valor de p. A rede de exemplo da Figura 2.4 tem $\langle C \rangle = 0,036$ (0 é uma rede sem aglomeração e 1 é o valor para uma rede aglomerada).

Outra propriedade importante para o estudo de redes é o cálculo do caminho mínimo médio. Sejam dois vértices v_i e v_j de uma rede, existe um caminho entre eles se existir pelo menos uma sequência de vértices $\{v_i, v_i+1, v_i+2, ...,v_j\}$, em que v_k está conectado com v_k+1, sendo $i \leq k \leq j$. A sequência de arestas que ligam esses vértices do extremo v_i ao extremo v_j é chamada de caminho entre os vértices e o caminho mínimo é o menor caminho ℓ dentre todos os caminhos possíveis. A média dos menores caminhos entre todos os nós da rede é o caminho mínimo médio $\langle \ell \rangle$ da rede. Este índice representa, em média, qual o menor caminho entre dois nós quaisquer da rede e pode ser calculado pela Equação 2.5, onde ℓ_{ij} representa algum caminho que conecte os dois vértices.

$$\langle \ell \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} \ell_{ij} \tag{2.5}$$

Como exemplo, considere ainda a rede da Figura 2.4, que possui caminho mínimo médio $\langle \ell \rangle = 3,062$. Por fim, o último índice de redes importante para o entedimento deste trabalho é o diâmetro D, que é valor do maior caminho mínimo da rede.

Hoje se sabe que o modelo ER não é o mais adequado para representar a grande maioria dos sistemas reais encontrados na natureza, sociologia etc. Porém, seu estudo serve como ponto de partida para realizar análises e comparações. Outras topologias de redes complexas mais próximas às redes reais serão apresentadas nas subseções seguintes.

2.2.2 Modelo mundo pequeno

Por muitos anos, o modelo aleatório de Erdös-Rényi permaneceu como a única forma de representar problemas associados a redes. Com o passar do tempo, viu-se que a topologia aleatória não explicava algumas questões inerentes a redes reais, como: o surgimento de vértices muito mais conectados que outros - chamados de *hubs* - e incompatibilidade da distribuição de graus de algumas redes com a função de Poisson. As redes aleatórias possuem também um baixo coeficiente de aglomeração ou *clustering*. Diz-se que há aglomeração quando grupos de vértices possuem alta densidade de arestas entre si em relação aos demais vértices da rede. Outra característica das redes aleatórias é a alta distância média, que significa a quantidade média de arestas que é preciso percorrer para encontrar o caminho entre dois vértices quaisquer na rede.

O modelo de Watts e Strogatz (WATTS; STROGATZ, 1998) situa-se entre o grafo regular - alta aglomeração e alta distância média - e o grafo aleatório (baixa aglomeração e baixa distância média). O modelo WS apresenta as características descritas no estudo de Milgram (1967) sobre mundo pequeno, onde duzentos e noventa e seis indivíduos fo-

ram arbitrariamente selecionados em Nebraska e Boston e convidados a enviar cartas a conhecidos até chegar a uma pessoa-alvo, em Massachusetts. Ao final do estudo, sessenta e quatro cartas atingiram a pessoa-alvo. Dentro deste grupo, o número médio de intermediários entre o primeiro remetente e a pessoa-alvo foi de 5,2, cunhando a expressão "seis graus de separação".

O processo de criação de uma rede mundo pequeno, ou *small-world*, começa com uma rede regular com n nós, cada um conectado a seus K vizinhos por arestas não dirigidas. Cada aresta é, então, reconectada a outro vértice escolhido aleatoriamente, com uma probabilidade p, até que se complete uma volta no grafo. Depois o procedimento é repetido para cada vértice, considerando as arestas de seus segundos vizinhos, com a mesma probabilidade p. Isso é feito até que cada aresta do grafo regular tenha sido considerada uma vez (WATTS; STROGATZ, 1998). A Figura 2.6 ilustra esse processo de construção, evidenciando as características intermediárias, entre a regularidade (quando $p = 0$) e a aleatoriedade (quando $p = 1$).

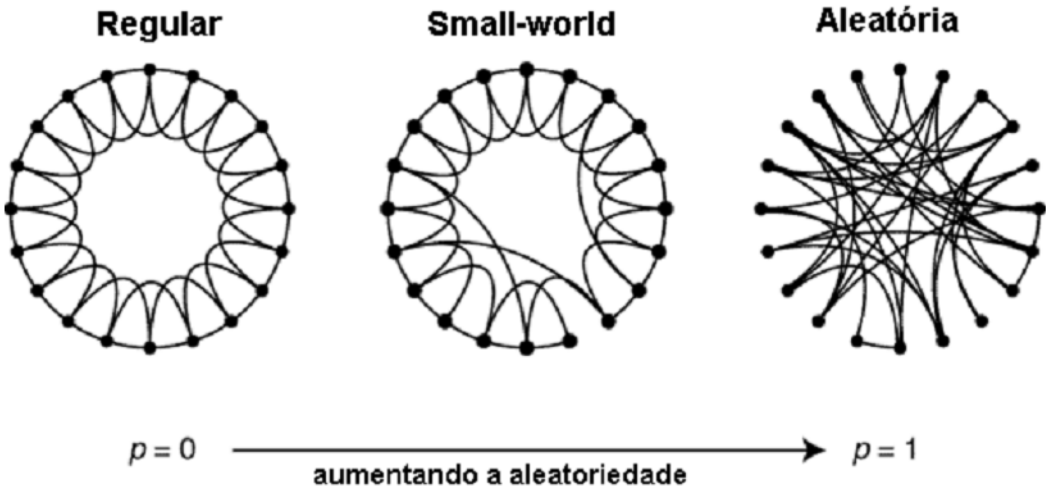

Figura 2.6: Processo de construção de uma rede mundo pequeno. Fonte: Adaptada de Watts e Strogatz (1998).

O resultado encontrado para valores intermediários de p $(0 < p < 1)$ é uma rede com a característica de alto coeficiente de aglomeração, como em redes regulares, e baixa distância média, como em redes aleatórias. O grau médio para o modelo WS é dado por $z = k = 2K$ e a distribuição de graus é similar ao modelo ER, conforme equação 2.2. As redes sociais são o principal exemplo de redes com topologia *small-word*.

2.2.3 *Modelo livre de escala*

A principal motivação para a criação do modelo livre de escala - *scale-free*, em Inglês - está na observação de algumas redes encontradas na natureza. Segundo preconizam

Barabási e Albert (1999), o tamanho de algumas redes e a complexidade das interações entre seus elementos impedem o conhecimento de suas verdadeiras topologias. Ainda segundo os pesquisadores, a ausência de dados reais sobre redes de larga escala impossibilita a validação do modelo aleatório de Erdös e Rényi (1959) no mundo real. Porém, com a utilização de computadores para automatizar a aquisição desses dados, os estudos da topologia e da dinâmica de redes do mundo real tornaram-se possíveis. A partir de então, observou-se a existência de um alto grau de auto-organização e, portanto, de complexidade, acompanhando as propriedades dessas redes.

Estudos (BARABÁSI; ALBERT, 1999; REDNER, 1998) mostraram que, independentemente do sistema e da natureza de suas interações, a probabilidade $P(k)$ de que um vértice na rede interaja com outros k vértices segue uma lei de potência denotada por: $P(k) \propto k^{-\gamma}$, conforme ilustra a Figura 2.7, que faz uma comparação com a distribuição normal. Esse modelo, denominado pela comunidade de modelo BA (Barabási e Albert), enfatiza não só as propriedades das ligações entre os vértices, mas também, o aspecto crescimento da rede.

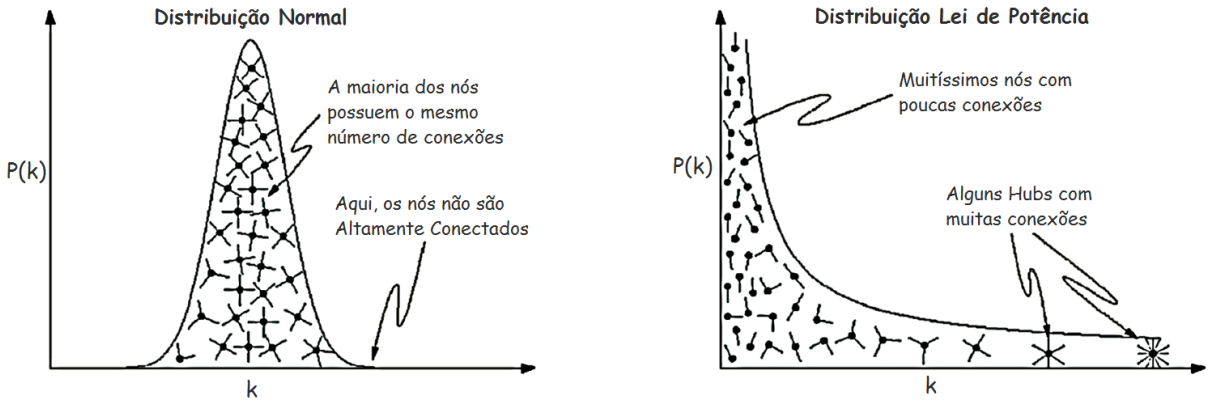

Figura 2.7: As redes aleatórias possuem uma distribuição de graus que segue uma curva normal (à esquerda), com um grau médio representativo; nas redes livres de escala a distribuição dos graus segue uma Lei de Potência (à direita). Fonte: Adaptada de Barabási (2007).

Ou seja, diferentemente dos modelos aleatório e mundo-pequeno, a rede não tem um número fixo de vértices. No modelo BA, a rede começa com um número pequeno de vértices m_0 e, a cada iteração de tempo é adicionado um novo vértice conectado através de m arestas aos vértices já existentes na rede, com $m \leq m_0$. Porém, as novas arestas não são conectadas de forma aleatória, mas sim, através da propriedade de ligação preferencial, que prevê: a probabilidade $\prod$ com que um vértice introduzido é conectado a outro vértice i depende do grau k_i que o vértice já possui, tal que:

$$\prod = \frac{k_i}{\sum_j k_j} \tag{2.6}$$

Após t passos, a rede possui $n = t + m_0$ vértices e $v = mt$ arestas. De acordo com a literatura especializada, as simulações realizadas em algumas redes reais de larga escala encontram um coeficiente para a lei de potência próximo de 3: $\gamma_{BA} = 3$ (BARABÁSI; ALBERT, 1999). A Figura 2.8 apresenta os gráficos e os respectivos valores para o coeficiente γ_{BA}, para três redes de larga escala estudadas: (A) Rede de colaboração de atores em filmes; (B) Rede WWW, formada pelas páginas Web conectadas por links entre os documentos; e (C) Rede de distribuição elétrica dos Estados Unidos.

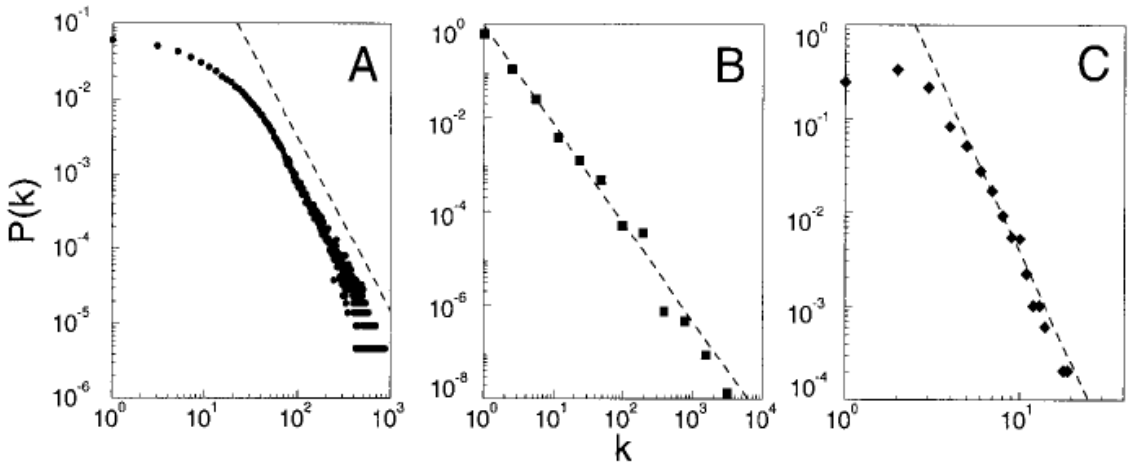

Figura 2.8: Função de distribuição de graus para diferentes redes de larga escala. (A) Gráfico de colaboração de atores com $n = 212.250$ vértices e grau médio $\langle k \rangle = 28{,}78$. (B) WWW, com $n = 325.729$ e $\langle k \rangle = 5{,}46$ (6). (C) dados da rede elétrica dos Estados Unidos, com $n = 4.941$ e $\langle k \rangle = 2{,}67$. As linhas tracejadas têm declives (A) 2,3, (B) 2.1 e (C) 4. Fonte: (BARABÁSI; ALBERT, 1999).

Assim, ao observar o modelo de crescimento apresentado acima, percebe-se a emergência de duas importantes propriedades das redes livres de escala: o fenômeno *rich-gets-riches*, ou os ricos ficam mais ricos, em Português, pois os nós mais conectados são os que recebem mais conexões; e o aparecimento dos *hubs*, ou nós altamente conectados. Os *hubs* assumem um papel fundamental no que tange à capacidade da rede de tolerância a erros ou ataques, propriedade também chamada de resiliência da rede, conforme indicam pesquisas na área (COHEN et al., 2000; ALBERT; JEONG; BARABÁSI, 2000; MOTTER, 2004).

O presente capítulo discorreu acerca das descobertas que levaram ao desenvolvimento da ciência das redes e suas aplicações na modelagem de estruturas existentes na natureza e no dia-a-dia das pesssoas. Tal apanhado é fundamental para o entendimento de um dos assuntos de que trata este documento. A outra área de estudo abordada por este trabalho é a Web Semântica, extensão da World Wide Web que será explorada com objetivo de gerar modelos de redes que auxiliem na compreensão de determinado domínio de estudo.

Alguns trabalhos publicados pela comunidade científica cuidam de mapear a World Wide

Web para um modelo de rede complexa, a fim de catalogar sua topologia e mensurar seu diâmetro. Em Albert, Jeong e Barabási (1999) e Barabási, Albert e Jeong (2000), os autores construíram um robô para coletar os documentos e links de uma pequena porção da Web (pois seu tamanho total estimado é de 8×10^8 documentos, o que torna a construção do modelo completo tecnologicamente inviável). O domínio `nd.edu.domain` foi utilizado para gerar o grafo direcionado, de onde emergiu a propriedade de lei de potência, com coeficientes $\gamma_{saída} = 2,45$ e $\gamma_{entrada} = 2,1$. Os pesquisadores também calcularam o menor caminho médio entre dois nós da rede, chegando à fórmula genérica $< d >= 0,35 + 2,06log(N)$. Com $N = 325.729$ nós, o valor de $< d >$ para o modelo extraído pelos pesquisadores é de 11,6 e, transportado para os $N = 8 \times 10^8$ da Web completa, obtiveram o índice $< d_{web} >= 18,59$. Isso significa que a menor distância entre dois documentos quaisquer na Web é de aproximadamente 19, i.e. apesar do gigantesco tamanho da rede, a Web é um grafo altamente conectado, o que facilita a ação de motores de busca inteligentes. Em contrapartida, isso torna a rede pouco resiliente a ataques, se direcionados a alguns pares de *hubs* fortemente conectados (ALBERT; JEONG; BARABÁSI, 2000).

O presente trabalho diferencia-se por utilizar os documentos da Web Semântica para gerar o modelo, ao invés da World Wide Web. Na Web Semântica, os elementos possuem identificadores e são ligados a outros através de prorpiedades ou predicados, ao contrário de *links*. O estado da arte da Web Semântica será discutido no próximo capítulo.

Capítulo Três

Web Semântica

Este capítulo visa apresentar os conceitos envolvidos nas definições da Web Semântica, enquanto extensão da World Wide Web. A primeira seção conta um breve histórico desde o surgimento da Web até o desenvolvimento das tecnologias da Web Semântica. Já a segunda seção explora as principais tecnologias citadas na seção anterior.

3.1 Breve Histórico

Segundo a recomendação do World Wide Web Consortium (W3C) (W3C, 2004), a World Wide Web (WWW, ou simplesmente Web) é um espaço de compartilhamento de informação que utiliza a Internet como plataforma e onde os recursos disponíveis são identificados globalmente através de *Uniform Resource Identifier* (URI). O tipo mais comum de URI é o *Uniform Resource Locator* (URL), usado para identificar endereços de páginas Web (BERNERS-LEE; HENDLER; LASSILA, 2001). Uma forma simplificada de descrever sua arquitetura é dizer que a Web possui três pilares: recurso, identificação e representação, enumerados a seguir:

a. Os **recursos** são os documentos compartilhados, chamados comumente de páginas Web;

b. Cada recurso possui um **identificador** único, URI, através do qual se obtém acesso ao seu conteúdo, utilizando um navegador e;

c. **Representação** é a forma como o conteúdo dos documentos é estruturado, através do uso de linguagens de marcação e estilo, como *HyperText Markup Language* (HTML) ou o *Resource Description Framework* (RDF), que será abordado mais adiante.

Para ilustrar essa arquitetura, a Figura 3.1 apresenta um exemplo onde o usuário deseja obter informações sobre a linguagem Python na Web.

A World Wide Web foi inventada em 1989 por Tim Berners-Lee como uma forma de compartilhar documentos entre pesquisadores. O grande diferencial do seu projeto está no uso de hipertexto, que consiste em adicionar ligações (*hiperlinks* ou apenas *links*) entre os documentos, criando uma forma de navegação não-linear entre eles. Podemos dizer que a Web transformou a Internet em uma rede estruturada de informações de alcance global.

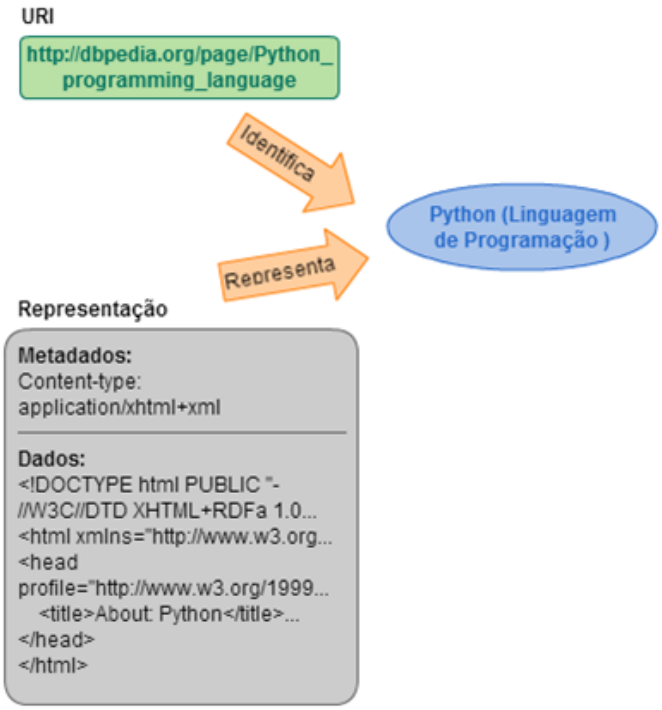

Figura 3.1: Relacionamento entre identificador, recurso e representação. Fonte: Adaptado de W3C (2004).

Desde então, a Web sofreu diversas alterações, não tanto em sua arquitetura original, mas principalmente na sua utilização, o que permitiu a evolução gradativa dos seus documentos, antes estáticos, para uma plataforma de serviços onde a abordagem principal é de colaboração. Em outras palavras, a responsabilidade pela construção do conteúdo passou dos desenvolvedores para os usuários da rede que, diariamente, alimentam *blogs*, redes sociais e outras plataformas com toda sorte de informação. Se, por um lado, esta nova característica da Web fez com que o volume de conteúdo publicado crescesse exponencialmente, por outro, fez também com que a Internet se tornasse um repositório desordenado de dados, dificultando a comunicação entre as aplicações da Web. O idealizador e criador da WWW - Tim Berners-Lee - percebeu este aspecto e, desde 2001, empreende esforços no sentido de classificar o conteúdo da Web de forma a imprimir semântica às palavras dentro de seus contextos.

Surgiu, então, a ideia de Web Semântica, cujo marco fora a publicação de um artigo na revista Scientific American, intitulado: "Web Semântica: um novo formato de conteúdo para a Web que tem significado para computadores vai iniciar uma revolução de novas possibilidade" (BERNERS-LEE; HENDLER; LASSILA, 2001), em conjunto com outros dois pesquisadores: James Hendler e Ora Lassila. A partir de então, estudiosos da área desenvolvem pesquisas e disseminam os conceitos da Web Semântica por meio da comunidade científica W3C Semantic Web Activity, que faz parte do W3C.

Os principais fatores de motivação apontados pela comunidade para o desenvolvimento da Web Semântica são: a conectividade das aplicações e a portabilidade de dados. Ou seja, seu objetivo é fazer com que as aplicações falem uma "língua" em comum e, portanto, possam se comunicar através da troca de informações semânticas. As pesquisas neste campo indicam a Web Semântica como uma nova geração da WWW que, assim como a Web revolucionou os meios como conectamos e consumimos documentos, revolucionará o modo como descobrimos, acessamos, integramos e usamos dados (HEATH; BIZER; CHRISTIAN, 2011).

3.2 Tecnologias da Web Semântica

Embora muitos sites da Web possuam algum grau de estrutura, a linguagem em que são criados, *Hypertext Markup Language* (HTML), é orientada para a estruturação de documentos textuais e não de dados. Para que as páginas da Web forneçam dados semi-estruturados, é preciso adicionar a elas marcações semânticas, chamadas de metadados, que identificam unicamente os recursos nelas publicados e seus relacionamentos com outros recursos do mesmo documento ou de outros.

Metadados são informações sobre as informações. Em uma biblioteca, por exemplo, estas marcações ajudam na localização de um livro, que foi catalogado previamente de acordo com o autor, assunto etc. A Web se parece um pouco com uma grande biblioteca de todo tipo de informação, que podem ser acessadas se você souber a URI (BRAY, 1998). Sem metadados, pode-se realizar buscas convencionais na Web apenas por palavras-chave, conforme dito anteriormente. Porém, se esta mesma Web for alimentada com metadados, a recuperação de informações pode levar em conta o contexto - ou semântica - de interesse do usuário e retornar um resultado mais satisfatório.

Esses metadados são providos com uso das tecnologias da Web Semântica, também chamados de microformatos[1]. Esta seção tem o objetivo de apresentar as principais delas.

3.2.1 RDF

O padrão recomendado pelo W3C para representação de metadados é o *Resource Description Framework* (RDF). Segundo W3C (2004), o RDF é um *framework* comum para expressar informações sobre recursos na Web, de modo que tais informações possam ser trocadas entre as aplicações, sem perda de sentido. Por ser um *framework* comum, endossado pelo W3C, torna-se uma motivação para que os projetistas de aplicações Web

[1] *Microformats*: http://microformats.org

desenvolvam padrões para aplicação de RDF na construção de suas páginas.

Em RDF, os elementos do mundo real são representados por triplas de sujeito, predicado (ou verbo) e objeto. Tomando como exemplo o contexto"linguagens de programação", a Figura 3.2 ilustra a declaração: "Python é uma linguagem de programação, criada em 1991", em forma de grafo.

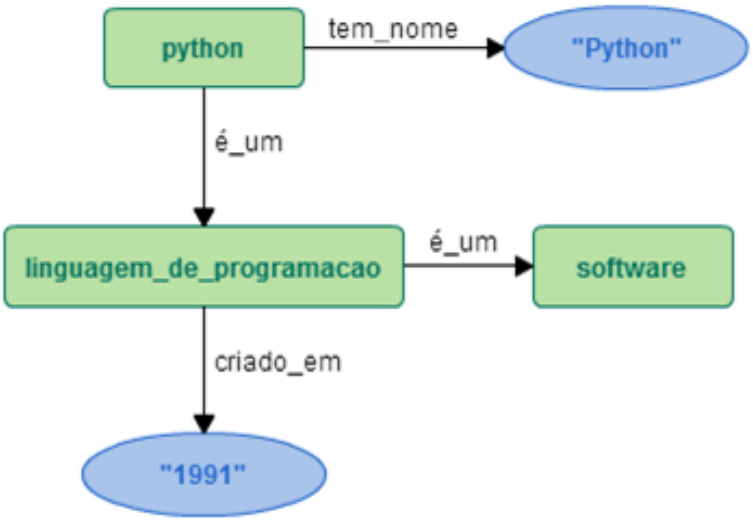

Figura 3.2: Exemplo de uma afirmação RDF em forma de grafo. Fonte: Elaborada pela autora.

Neste tipo de analogia formada por vértice-aresta-vértice, o sujeito é o elemento que está no vértice inicial, o predicado é representado pelo nome da aresta e o objeto é o elemento do vértice final. No exemplo, a aresta que vai do vértice "python" até o vértice "linguagem_de_programação", chamada "é_um", forma a tripla que representa a afirmação "Python é uma linguagem de programação". Em RDF, sujeitos, predicados e objetos são nomes de entidades, também chamadas de recursos. Entidades podem representar coisas do mundo real - como um *software* - ou algo mais abstrato, como estados e relações (TAUBERER, 2006).

O padrão W3C para codificar RDF é a linguagem de marcação *eXtensible Markup Language* (XML). A Figura 3.3 ilustra o trecho de um arquivo RDF codificado em XML, correspondente ao exemplo da Figura 3.2. Nela, é possível observar que cada entidade ou recurso é representado por um nome, ou URI, que o identifica unicamente, conforme apresentado anteriormente. Ou seja, tanto os sujeitos como os verbos e os objetos têm seu identificador único. Além disso, objetos podem ser representados por literais, como "Python", que é o nome da linguagem de programação, ou "1991", que é o ano em que ela foi criada.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                 xmlns:foaf="http://xmlns.com/foaf/0.1/"
                 xmlns:dbpprop="http://dbpedia.org/property>
        <rdf:Description rdf:about="http://dbpedia.org/resource/Python_(programming_language)">
                <rdf:type rdf:resource="http://dbpedia.org/ontology/ProgrammingLanguage"/>
                <foaf:name xml:lang="en">Python</foaf:name>
                <dbpprop:year rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1991</dbpprop:year>
        </rdf:Description>
</rdf:RDF>
```

Figura 3.3: Exemplo de uma afirmação RDF codificada em XML. Fonte: Elaborada pela autora.

3.2.2 Linked Data

Segundo Heath, Bizer e Christian (2011), a tecnologia Linked Data provê um paradigma de publicação de onde não apenas documentos, mas também dados, podem ser "protagonistas" da Web. Os mesmos autores definem a Web Semântica como uma Web de dados (*Web of data*, em Inglês). Trata-se de uma extensão da Web convencional, que a transforma em um espaço global de dados, baseado em padrões abertos. O fato do Linked Data utilizar padrões abertos é a principal característica que permite a interoperabilidade de dados e, consequentemente, sistemas, nessa plataforma.

Por um lado, a cada dia um número crescente de indivíduos e empresas escolhe compartilhar seus dados na Web. Por outro lado, há empresas e comunidades científicas interessadas em consumir esses dados, como, por exemplo, as empresas desenvolvedoras de motores de busca, pois, além de melhorar a experiência do usuário, se gasta menos recursos para pesquisar dados semiestruturados, ao invés de simples páginas HTML. Outro exemplo de parte interessada em Linked Data está no campo das pesquisas científicas: disciplinas ligadas às Ciências da Vida se beneficiam fortemente quando há intercâmbio mundial de dados de pesquisa entre cientistas, como demonstrado pelo progresso resultante de iniciativas de cooperação, como o Projeto Genoma Humano (HEATH; BIZER; CHRISTIAN, 2011).

De acordo com Berners-Lee (2001), Linked Data é um conjunto de princípios e tecnologias que aproveita a própria arquitetura da World Wide Web para viabilizar o compartilhamento e reuso de dados em grande escala. O projeto *Linking Open Data* (LOD)[2] é mantido por um grupo de trabalho do W3C. Ele disponibiliza uma representação visual de seus *datasets* em forma de diagrama, onde cada base é representada por um círculo e há *links* entre eles. A Figura 3.4 apresenta a versão colorida do diagrama, onde as cores servem para agrupar as bases do mesmo tema.

[2] http://linkeddata.org

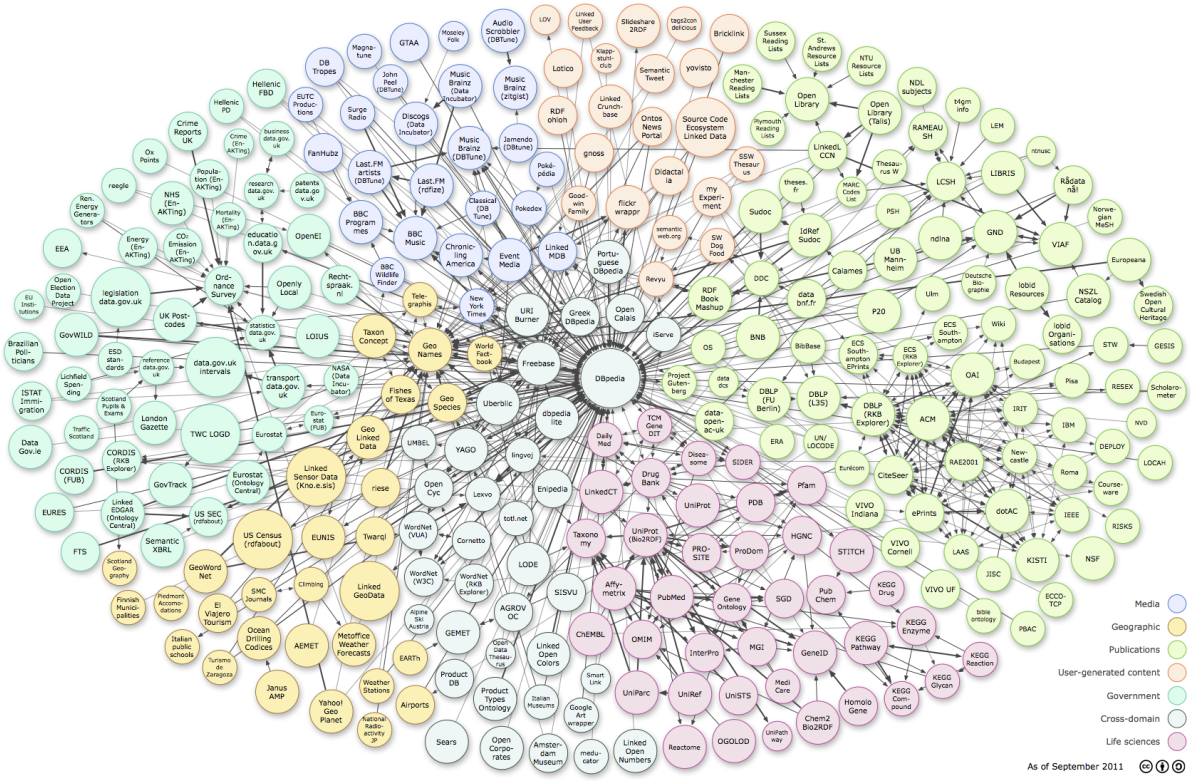

Figura 3.4: Linking Open Data cloud diagram, por Richard Cyganiake e Anja Jentzsch. Fonte: http://lod-cloud.net/.

Ao clicar em um *dataset*, o leitor é redirecionado para a página do projeto correspondente. As bases de dados do projeto LOD são escritas em RDF e podem ser consultas através da linguagem SPARQL, que será elucidada na próxima subseção.

3.2.3 SPARQL

Em um ambiente estruturado, a SQL (*Structured Query Language*, em Inglês, ou Linguagem de Consulta Estruturada) é o padrão mais conhecido no mercado para acessar bases de dados relacionais. A expressão "relacional" advém das características de álgebra relacional nas quais a SQL foi inspirada. A linguagem correspondente para acessar bases de dados semiestruturadas é chamada de SPARQL, um acrônimo recursivo para *SPARQL Protocol and RDF Query Language*, i.e. Protocolo e Linguagem de Consulta SPARQL. É recomendada pelo W3C como a linguagem de consulta para bases de dados RDF e, por isso, considerada pela comunidade científica como a linguagem de consulta da Web Semântica.

A linguagem SPARQL permite realizar desde consultas mais simples até explorar relações desconhecidas, através de junções em bases RDF diferentes (W3C, 2013a). Uma consulta SPARQL tem o formato de tripla, assim como a linguagem RDF, com a diferença de que cada elemento - sujeito, predicado e objeto - pode ser uma variável. A sintaxe de uma consulta SPARQL compreende os seguintes elementos:

a. **Declarações de prefixos**, que são caminhos relativos ou nomes para abreviar os valores absolutos das URIs;

b. **Definição do conjunto de dados (*dataset*)**, para informar em qual base RDF os dados serão consultados;

c. **Cláusula resultado**, que identifica quais informações serão retornadas a partir da consulta;

d. **Padrão de consulta**, que especifica o formato ao qual os resultados devem corresponder;

e. **Modificadores de consulta**, para ordenar, limitar ou organizar os resultados.

A Figura 3.5 ilustra a organização de cada elemento citado acima, na consulta.

Os prefixos são declarados utilizando-se a cláusula PREFIX (linha 2). No exemplo acima, a URI "http://exemplo.com/recursos" foi abreviada pelo termo "ex". Isso significa que a expressão: "http://exemplo.com/recursos#umRecurso" pode ser acessada na consulta

```
1  #1. declarações de prefixos
2  PREFIX ex: <http://exemplo.com/recursos/>
3  ...
4  #2. definição do conjunto de dados
5  FROM ...
6  #3. cláusula resultado
7  SELECT ...
8  #4. padrão de consulta
9  WHERE{
10     ...
11 }
12 #5. modificadores de consulta
13 ORDER BY...
```

Figura 3.5: Sintaxe SPARQL. Fonte: Adaptada de Feigenbaum e Prud'hommeaux (2013).

por: "ex:umRecurso". Os conjuntos de dados ou datasets são elencados na cláusula FROM (linha 5). Já os resultados, padrões de consulta e modificadores são representados pelas cláusulas SELECT (linha 7), WHERE (linha 9) e outras cláusulas chamadas modificadoras de solução (linhas 13 em diante), respectivamente. A Figura 3.6 traz um exemplo de consulta SPARQL executada contra uma base RDF remota.

```
1  PREFIX dbpedia: <http://dbpedia.org/ontology/>
2  PREFIX dbprop: <http://dbpedia.org/property/>
3  SELECT ?lang
4  WHERE {
5      ?lang a dbpedia:ProgrammingLanguage .
6      ?lang dbprop:year 1991.
7  }
```

Figura 3.6: Consulta SPARQL que retorna as linguagens de programação criadas em 1991. Fonte: Elaborada pela autora.

Nesse exemplo foram utilizados dois apelidos para URIs: "dbpedia" e "dbprop", declarados na cláusula PREFIX, linhas 1 e 2. A cláusula WHERE define o padrão de grafo básico, que deverá corresponder a um subgrafo do conjunto de dados RDF definido na cláusula FROM. Isso significa que, ao substituir os elementos do grafo padrão pelos valores das variáveis, o grafo resultante é equivalente a um subgrafo do *dataset*.

A palavra-chave "a" pode ser usada como predicado da sentença, substituindo o relacionamento "rdf:type", representado pela URI: "http://www.w3.org/1999/02/22-rdf-syntax-ns#type", conforme aparece na linha 5: "?lang a dbpedia:ProgrammingLanguage". Esta sentença indica que a URI representada pela variável "?lang" tem um relacionamento "rdf:type" com "dbpedia:ProgrammingLanguage", o que significa que "?lang" é do tipo (é uma) linguagem de programação[3].

[3]Outros exemplos de consultas SPARQL podem ser encontrados em W3C (2013a).

As ferramentas que aceitam consultas via protocolo SPARQL e retornam resultados via protocolo HTTP são chamadas de *endpoints*. Estes podem ser genéricos, quando são capazes de consultar qualquer *dataset* RDF acessível pela Web, ou específicos, quando permitem a realização de consultas em *datasets* particulares. Utilizamos no presente trabalho, para os exemplos e experimentos, o *endpoint* específico: Virtuoso SPARQL[4], que permite consultas na base de dados DBpedia, *dataset* que reúne as informações contidas nas *infoboxes* das páginas da Wikipedia. A base de dados DBpedia será vista com detalhes no capítulo 4, onde são apresentados os experimentos de validação do modelo. Ao executar a consulta ilustrada na Figura 3.6 contra o *endpoint* Virtuoso SPARQL, tem-se o resultado representado na Figura 3.7.

lang
http://dbpedia.org/resource/Object-Oriented_Turing
http://dbpedia.org/resource/Oriel_(scripting_language)
http://dbpedia.org/resource/Haskell_(programming_language)
http://dbpedia.org/resource/Python_(programming_language)
http://dbpedia.org/resource/NewLISP
http://dbpedia.org/resource/Oz_(programming_language)
http://dbpedia.org/resource/Gemstone_(database)
http://dbpedia.org/resource/AgentSheets
http://dbpedia.org/resource/Oberon-2_(programming_language)
http://dbpedia.org/resource/Component_Pascal
http://dbpedia.org/resource/GNU_E
http://dbpedia.org/resource/QBasic

Figura 3.7: Resultado da consulta da Figura 3.6, utilizando o *endpoint* Virtuoso SPARQL.

Por fim, convém salientar que SPARQL é uma linguagem e também um protocolo, conforme anuncia seu acrônimo *SPARQL Protocol and RDF Query Language*. O protocolo SPARQL foi baseado no protocolo HTTP (*Hypertext Transfer Protocol*) e padroniza as regras e formatos para envio de consultas e recebimento de resultados. Funciona como um *Web Service*, que deve ser implementado por todos os *endpoints* SPARQL.

De acordo com W3C (2013b), uma consulta simples é enviada via método GET, utilizando o parâmetro "query" do protocolo. Já uma consulta maior - comumente aquelas geradas por agentes automatizados - pode ser enviada via método POST. A sequência de três figuras a seguir exemplifica de forma resumida algumas definições do protocolo SPARQL para conversão de *queries* em serviços de processamento de *queries*. Na Figura 3.8 tem-se uma *query* simples em SPARQL, utilizando a base RDF Dublin Core[5].

[4] http://dbpedia.org/sparql
[5] http://dublincore.org/

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?book ?who
WHERE { ?book dc:creator ?who }
```

Figura 3.8: Exemplo de *query* SPARQL simples. Fonte: (W3C, 2013b).

Ao executar a *query* contra um *endpoint*, esta é convertida para o formato HTTP, conforme a Figura 3.9, que esquematiza a operação GET. Uma abstração foi feita ao codificar a *query* da Figura 3.8 na string "EncodedQuery".

```
GET /sparql/?query=EncodedQuery
Host: www.example
User-agent: my-sparql-client/0.1
```

Figura 3.9: *Query* anterior convertida para o protocolo HTTP. Fonte: Adaptada de W3C (2013b).

O retorno da consulta pode ser definido usando o parâmetro "accept", que pode assumir diferentes formatos, como CSV, XML, JSON, RDF, HTML, *Turtle*, texto simples, dentre outros. Como o valor do parâmetro não foi explicitado nos exemplos, o resultado padrão é retornado em XML, conforme ilustra a Figura 3.10.

```
HTTP/1.1 200 OK
Date: Fri, 06 May 2005 20:55:12 GMT
Server: Apache/1.3.29 (Unix) PHP/4.3.4 DAV/1.0.3
Connection: close
Content-Type: application/sparql-results+xml

<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">

 <head>
   <variable name="book"/>
   <variable name="who"/>
 </head>
 <results>
   <result>
     <binding name="book"><uri>http://www.example/book/book5</uri></binding>
     <binding name="who"><bnode>r29392923r2922</bnode></binding>
   </result>
...
</sparql>
```

Figura 3.10: Resultado da consulta, em XML. Fonte: (W3C, 2013b).

Os conceitos apresentados até aqui são importantes para o entendimento do desenvolvimento do presente projeto de pesquisa. A partir do capítulo subsequente, o projeto será apresentado em todas as etapas de seu desenvolvimento, culminando nos resultados e

discussões acerca de um experimento de validação do modelo proposto por este trabalho.

Capítulo Quatro

Modelo proposto

Este capítulo tem início com um relato sobre o caminho metodológico percorrido para se alcançar os objetivos almejados por este trabalho. Em seguida, apresenta-se a proposta de um modelo computacional, que é o cerne da pesquisa em questão e alguns trabalhos relacionados, como forma de comparação. Para validação do modelo, apresenta-se um experimento que utiliza parâmetros para análise da eficiência do modelo, dentro do cenário a que se propõe. Por fim, é feita uma discussão acerca dos resultados encontrados.

A pesquisa teve início com o problema que envolve duas áreas de pesquisa distintas: Web Semântica e redes complexas. A Web Semântica é um repositório bastante rico de informações em todas as áreas do conhecimento, que se encontra em constante expansão. Como este trabalho adota a premissa de que a Web Semântica é também um conjunto de redes semânticas, surgiu a motivação para estudar essas bases de dados utilizando índices e características da teoria de redes.

A necessidade de investigar mais a fundo os temas que compreendem o problema levaram à formação de um grupo de pesquisa, com os participantes: (i) Eduardo Manuel de Freitas Jorge, Doutor em Difusão do Conhecimento no programa multi institucional pela UFBA\LNCC\UNEB\UEFS / IFBA\SENAI-CIMATEC, Professor da Universidade do Estado da Bahia (UNEB); (ii) Gabriela Oliveira Mota da Silva, mestranda em Modelagem Computacional e Tecnologia Industrial, pelo SENAI CIMATEC; (iii) Hernane Borges de Barros Pereira, Doutor em Engenharia Multimídia pela Universitat Politècnica de Catalunya (UPC) e Professor do SENAI ClMATEC; (iv) ShankarCabus de Teive e Argollo, graduando do curso de bacharelado em Sistemas de Informação, pela Universidade do Estado da Bahia.

Assim, o trabalho iniciou-se com o levantamento bibliográfico dos assuntos: modelos de criação, organização, gestão e difusão do conhecimento; Web Semântica; RDF e Linked Data; SPARQL; redes semânticas; redes complexas; ferramentas para análise de redes. Após obter o embasamento teórico, o objetivo do grupo de pesquisa pôde ser definido: propor um modelo computacional para contribuir com a análise de modelos de criação, organização, gestão e difusão do conhecimento, dentro do contexto da Web Semântica, sob a perspectiva da teoria das redes sociais e complexas. Para alcançar tal objetivo, foi necessário trabalhar com protocolos de conversão de dados de diferentes fontes, para formatos legíveis por ferramentas de análise de redes.

Com base nesse conjunto de definições e em algumas delimitações do escopo, foi descrita

uma arquitetura inicial para o modelo, contendo um protocolo para coleta de dados baseado em RDF e um protocolo de conversão dos elementos coletados para um formato de rede, conforme Figura 4.1.

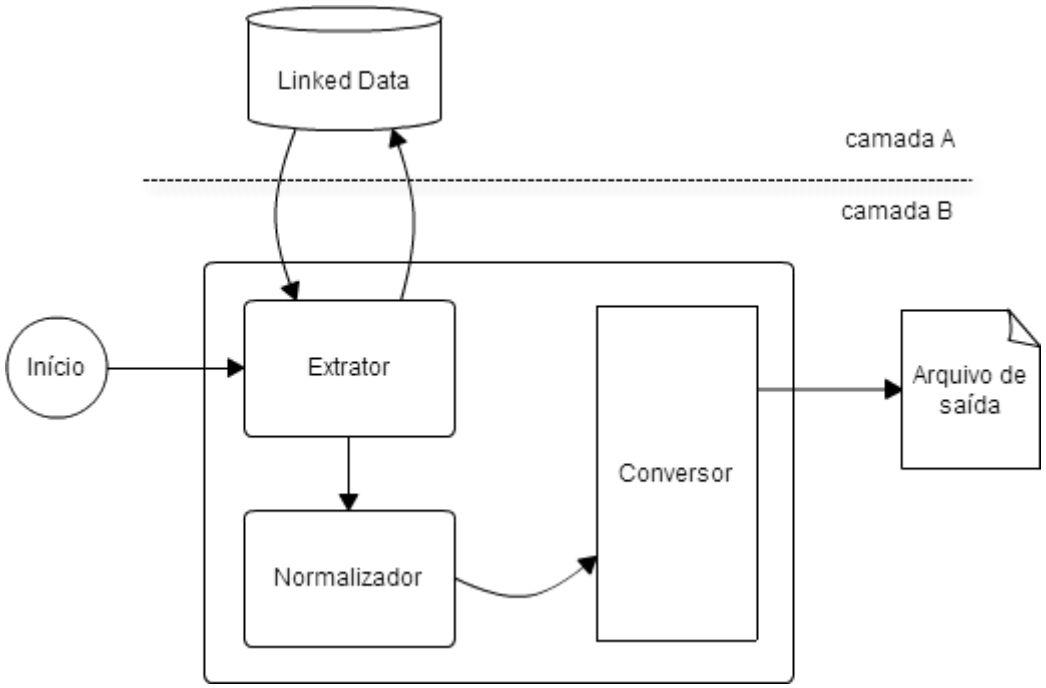

Figura 4.1: Arquitetura inicial do modelo. Fonte: (ARGOLLO, 2012).

A arquitetura descreve um modelo baseado em três módulos: Extrator, Normalizador e Conversor. O módulo Extrator se comunica com uma base de dados *Open Linked Data* para coletar insumos de um determinado domínio de estudo. Em seguida, o módulo Normalizador converte as triplas RDF em um formato padronizado, que servirá de entrada para o módulo Conversor, responsável por montar o arquivo de saída no formato de rede escolhido. Todo esse funcionamento será explicado com mais detalhes na próxima seção, onde será apresentada uma nova arquitetura evoluída a partir dos resultados dessa primeira definição do modelo.

Em 2012, o aluno de graduação Shankar Cabus de Teive e Argollo apresentou os primeiros resultados do grupo de pesquisa na defesa de sua monografia de conclusão do curso de bacharelado em Sistemas de Informação, pela Universidade do Estado da Bahia, intitulada: "Uma solução computacional para integração entre Web Semântica e redes complexas", que apresentou a primeira versão do modelo (ARGOLLO, 2012). Uma ferramenta *open-source* foi construída, como forma de validação do modelo proposto, e recebeu o nome de RDFree. Para verificar o modelo, Argollo (2012) sugeriu o domínio de estudo: bandas e gêneros musicais. O experimento consistiu na coleta dos dados RDF da base DBpedia, conversão dos dados para o formato GEXF, utilizando a ferramenta RDFree e, por fim, análise e caracterização topológica da rede a partir de cálculos dos índices mais relevantes para o domínio de estudo, utilizando a ferramenta GEPHI.

Do trabalho supracitado, concluiu-se que o modelo é promissor, pois alcançou o objetivo de converter dados da Web Semântica para auxiliar pesquisadores especialistas em redes complexas. No presente trabalho, objetiva-se ampliar o RDFree, tornando o modelo mais robusto e, assim, atingir um público maior de pesquisadores interessados em análise de outros tipos de bases de dados e outros formatos de redes. A partir dessas conclusões, o modelo e a ferramenta RDFree ganham novos módulos e novos experimentos, a fim de validar a nova arquitetura. As próximas seções apresentam essa evolução, detalhando a sua definição - arquitetura e requisitos, os experimentos realizados, resultados e discussões a respeito.

4.1 Definição do modelo

4.1.1 Arquitetura

O modelo proposto foi especificado com base em três módulos independentes entre si, de acordo com o diagrama da arquitetura exibido na Figura 4.2. O pesquisador que deseja utilizar o modelo interage com uma engrenagem que coordena a operação dos módulos. As linhas pontilhadas na figura delimitam as fronteiras do modelo com o ambiente externo. A fronteira superior representa a entrada de dados coletados e a fronteira inferior representa a saída dos dados convertidos em arquivos de rede.

O fluxo de funcionamento do modelo se inicia com a ação do pesquisador de escolher o tipo de dado que se deseja trabalhar, pois para cada tipo é necessário que esteja implementado um protocolo de coleta. Os dados podem ser do tipo estruturado, como em bancos de dados relacionais. Neste caso, o protocolo de coleta deve utilizar a linguagem SQL para extrair os dados e seus relacionamentos relevantes para a pesquisa. Outro exemplo de tipo de dado são os dados semiestruturados, como aqueles presentes na arquitetura da Web Semântica: RDF, OWL etc. A linguagem oficial para consultas nessa grande base de dados online é a SPARQL, por isso o protocolo de extração para esse tipo de dado deve ser implementado nesta linguagem. Um último exemplo, dessa vez referente ao tipo de dado não estruturado, são os arquivos de texto, como os que possuem as extensões PDF, ODF etc. Para estes casos, a implementação do protocolo de coleta deve envolver mineração de dados, pois os mesmos não possuem qualquer relacionamento pré-estabelecido entre eles.

A partir da definição do protocolo de coleta, o módulo Coletor extrai os dados e repassa para o módulo Normalizador. Este módulo prepara os dados, utilizando um formato padrão, permitindo, assim, que os diferentes protocolos de conversão trabalhem sempre com o mesmo tipo de entrada. Isso garante a independência entre os módulos Coletor e Conversor. O protocolo de conversão também é definido a partir da escolha do pesquisa-

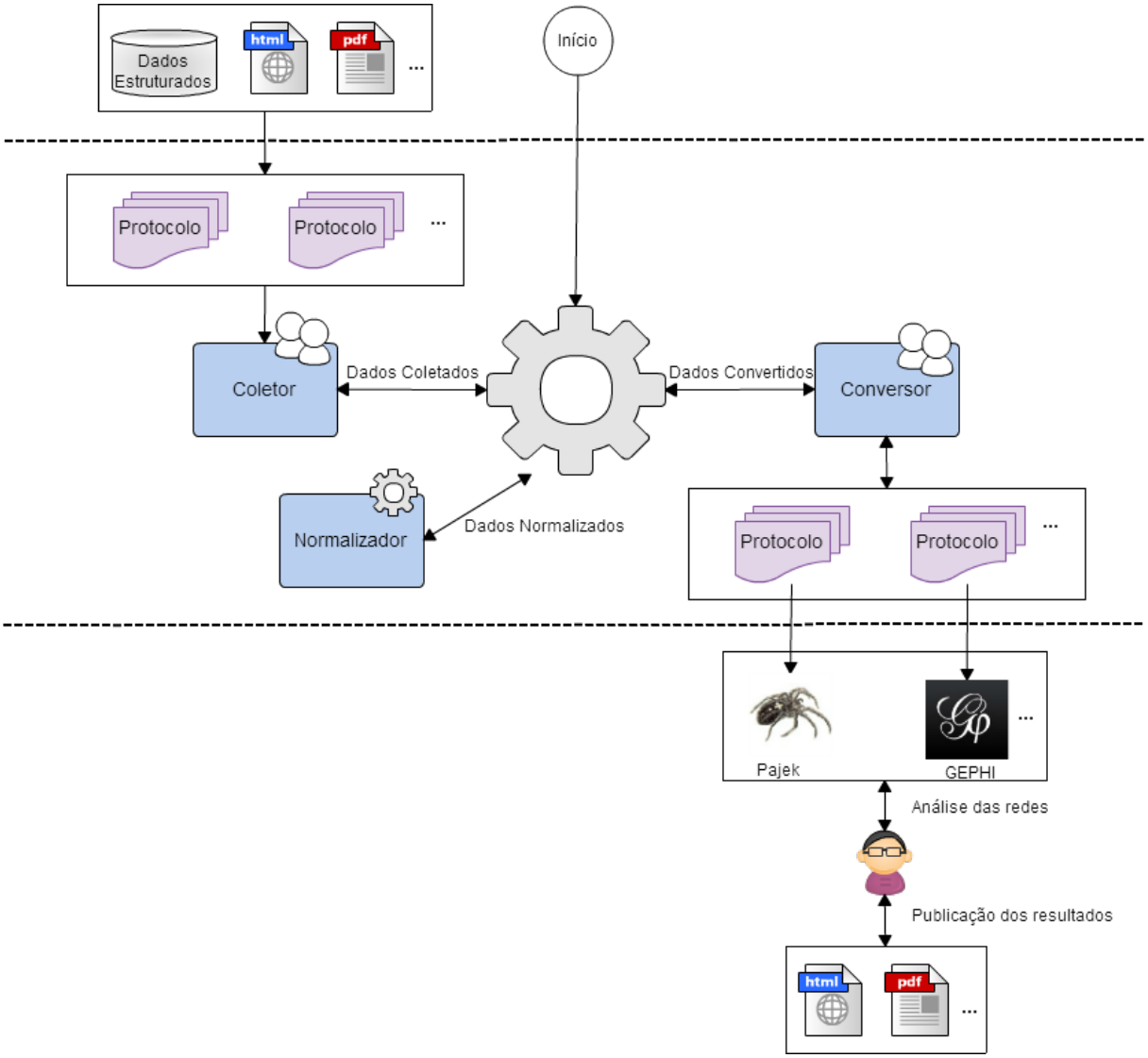

Figura 4.2: Arquitetura geral do modelo.

dor, a depender da ferramenta de redes que ele deseja utilizar para analisar os dados de sua pesquisa. Estes também precisam estar implementados previamente no modelo. Feito isso, o módulo Conversor vai utilizar o protocolo de conversão escolhido para transformar a entrada de dados de formato padrão para o formato de arquivo de redes escolhido pelo pesquisador.

Após essa breve explicação sobre o fluxo do modelo, a arquitetura será detalhada, através da descrição do funcionamento de cada módulo presente na Figura 4.2.

Coletor – este módulo é responsável por importar os dados a serem analisados pelo pesquisador. Para cada tipo de dado de entrada deve haver uma instância desse módulo capaz de obtê-los. Cada instância corresponde à implementação de um protocolo de coleta ou extração de dados. Por exemplo: se a entrada for do tipo de dado estruturado, como em um banco de dados relacional, a instância do Coletor correspondente deve implementar um protocolo que conecte-se à base de dados, execute uma dada consulta SQL e importe os resultados para o modelo. Ou, se a entrada for um documento não estruturado, em formato PDF, por exemplo, a instância do coletor deve conter a implementação de um algoritmo de mineração de texto como protocolo de extração. Assim, cada entrada do modelo terá um formato diferente, a depender da implementação de sua instância.

Normalizador – este módulo resolve o problema gerado pelas entradas de dados diferentes, ao prepará-los para que o módulo Conversor trabalhe de maneira independente. Neste módulo, é criada uma estrutura de dados simples, contendo os nós e arestas da futura rede, utilizando o formato JSON, que será explicado na seção 4.3.2. Esse passo é necessário, pois cada protocolo de conversão trabalhará de forma diferente, mas todos partirão do mesmo formato de arquivo normalizado, sendo pré-condição para que o modelo implemente a funcionalidade de conversão independentemente do tipo de dado de entrada do modelo.

Conversor – este módulo funciona como um tradutor dos dados normalizados para o formato da ferramenta de análise de redes na qual o pesquisador deseja analisar os dados. Para cada ferramenta de análise de redes, deve haver uma instância desse módulo capaz de gerar um arquivo na extensão correspondente. Ou seja, cada instância corresponde à implementação de um protocolo de conversão de dados para um formato de arquivo de redes. Ao executá-lo, os dados são convertidos e escritos em um arquivo, cujo formato dos nós e arestas é corretamente interpretado por uma ferramenta de rede, por exemplo: Pajek, Gephi etc. A partir de um mesmo conjunto de dados normalizado em formato JSON, é possível gerar tantos arquivos de redes quantas sejam as instâncias desse módulo implementadas.

Como resultado final do modelo, o pesquisador obtém uma rede complexa e, ao inter-

pretá-la com a ferramenta de análise de redes correspondente, o pesquisador é capaz de identificar visualmente os relacionamentos entre seus elementos, calcular os índices de redes, como grau médio, aglomeração etc. e, a partir daí, dissertar e extrair conclusões sobre o universo pesquisado. Os especialistas podem também publicar seus resultados em diferentes formatos, como HTML, PDF, Mapas Mentais ou outros, aumentando o alcance dos impactos gerados por sua pesquisa.

4.1.2 Requisitos funcionais e não funcionais

A fase de especificação dos requisitos funcionais e não funcionais visa garantir que tanto as funcionalidades relevantes para o modelo estejam contempladas por sua arquitetura, como também garantir os requisitos que não se referem especificamente a funcionalidades, mas a aspectos técnicos, como: qualidade, confiabilidade, flexibilidade e interoperabilidade do modelo. A Tabela 4.1 relaciona os requisitos funcionais do modelo, onde a primeira coluna representa o identificador do requisito e a segunda coluna, sua descrição.

Tabela 4.1: Requisitos funcionais do modelo.

ID	**Descrição**
RF1	O modelo deve importar dados estruturados, semiestruturados ou não estruturados de bases remotas ou locais.
RF2	Os dados devem ser preparados e normalizados para um formato padrão.
RF3	Os dados normalizados devem ser convertidos para um arquivo no formato de rede escolhido.

Como o objetivo da pesquisa é o uso do modelo por pesquisadores de diversas áreas do conhecimento, é necessário torná-lo flexível o bastante para permitir implementações de diferentes tipos de dados de entrada e saída. Para isso, alguns requisitos não funcionais foram definidos e, de forma similar à seção anterior, são relacionados na Tabela 4.2, onde a primeira coluna representa o identificador do requisito e a segunda coluna, sua descrição.

Tabela 4.2: Requisitos não funcionais do modelo.

ID	**Descrição**
RNF1	O modelo deve dar suporte à importação de novos tipos de bases de dados estruturadas, semiestruturadas e não estruturadas, através da inclusão de novos protocolos de coleta de dados.
	Continua na próxima página

Tabela 4.2 – Continuação

ID	Descrição
RNF2	O modelo deve dar suporte à conversão dos dados para novos formatos de arquivo de redes, através da inclusão de novos protocolos de conversão de dados.
RNF3	A partir dos dados convertidos pelo modelo, possibilitar que o pesquisador analise as propriedades da rede e publique os resultados em diferentes formatos (HTML, PDF etc.), com uso de ferramentas de análise de redes.

As especificações apresentadas nesta seção foram consolidadas na Tabela 4.3, onde são listados os módulos, sua descrição, elementos de entrada - input, elementos de saída - output. A última coluna da tabela faz a correspondência do módulo com os respectivos requisitos funcionais e não funcionais que ele implementa.

Tabela 4.3: Quadro-resumo da arquitetura geral do modelo.

Módulo	Descrição	Input	Output	Requisitos
Coletor	Responsável por se conectar à base de dados de qualquer tipo, executar um dado protocolo de conversão e importar os resultados para o modelo.	Base de dados e protocolo de coleta	Dados coletados	RF1 e RNF1
Normalizador	Executa os passos comuns a todos os protocolos de conversão, ou seja, prepara os dados para o módulo Conversor, criando uma estrutura de dados simples, contendo os nós e arestas da futura rede.	Dados coletados	Dados normalizados	RF2
Conversor	Funciona como um tradutor dos dados normalizados para o formato definido pelo protocolo de conversão escolhido ou vice-versa.	Dados normalizados e protocolo de conversão	Dados convertidos	RF3, RNF2 e RNF3

4.2 Trabalhos Correlatos

Durante a fase de definição do problema de pesquisa e dos objetivos deste trabalho, foram realizadas buscas na literatura atual para encontrar trabalhos relacionados, mas foram encontradas poucas pesquisas que relacionassem Web Semântica e redes complexas. Uma delas é o projeto *Living Semantic Web*[1], que tem o objetivo de modelar e analisar a Web Semântica como um sistema complexo. A comparação entre a Web Semântica e um sistema vivo - presente em seu título - faz alusão ao fato de que a maior parte dos sistemas modelados na literatura é composta por sistemas vivos, por exemplo: redes neurais e a cadeia alimentar, além de redes tecnológicas, como redes elétricas ou a World Wide Web.

O problema de pesquisa do projeto Living Semantic Web é resumido em: a Web Semântica pode ser modelada como um sistema vivo, que é uma subclasse de Sistemas Complexos? Segundo os pesquisadores, a resposta é sim, se considerado que ambos os sistemas apresentam as mesmas regularidades (previsibilidade, leis mecânicas) e dinâmica (auto-organização, adaptação ou evolução) (GIL; GARCÍA, 2005).

As conclusões dos pesquisadores foram baseadas no estudo de alguns índices de redes, presentes na figura : grau médio, coeficiente de agomeração, caminho mínimo médio e coeficiente de lei de potência. O estudo mostra que a porção da Web Semântica coletada apresenta características small-word, com alto índice de clusterização e, ao mesmo tempo, uma distribuição de graus livre de escala.

Tabela 4.4: Comparação entre as redes estudadas no projeto Living Semantic Web. Fonte: Adaptada de GIL e GARCÍA (2005)

Rede	**Nós**	K	C	ℓ	γ
DAMLOntos (11/04/2013)	56.592	4,63	0,152	4,37	-1,48
DAMLOntos (31/01/2005)	307.231	3,83	0,092	5,07	-1,19
CopyrightOnto	971	3,71	0,071	3,99	-3,29
WWW	∼200M	-	0,108	3,10	-2,10

Entretanto, o modelo Living Semantic Web possui a limitação de considerar apenas ontologias - mais especificamente a biblioteca de ontologias DAML (*DARPA*[2] *Agent Markup Language*), como forma de representar as bases da Web Semântica, ou seja, de modelar as propriedades e o comportamento dela própria como uma rede complexa. O presente trabalho se diferencia, principalmente, por utilizar metadados instanciados como modelos de criação, organização, gestão e difusão do conhecimento na construção de suas redes, ou

[1] *Living Semantic Web*: http://rhizomik.net/html/livingsw/
[2] http://www.darpa.mil/

seja, sistemas de representação do conhecimento que modelam redes reais. E, sendo redes reais, muitas delas já são consideradas como sistemas complexos, resultado de variados trabalhos científicos das últimas décadas.

Assim, é importante ressaltar que a presente pesquisa não tem o objetivo de demonstrar a natureza complexa da Web Semântica, mas sim, de prover meios para que outros sistemas e redes, representados através da Web Semântica, possam ser estudados e validados como sistemas e redes complexas.

Outro projeto correlato é o *Semantic Web Import*[3]: um *plugin* para a ferramenta de análise de redes Gephi que importa dados da Web Semântica através de *queries* SPARQL e os interpreta como grafos, exibindo na própria interface da ferramenta. O *plugin* permite aplicar consultas SPARQL tanto em arquivos RDF locais - no próprio computador do usuário, como também utiliza os protocolos SOAP (*Simple Object Access Protocol*) e REST (*REpresentational State Tranfer*) para realizar consultas em bases remotas, através de endpoints. Nesta modalidade, a ferramenta oferece ainda a possibilidade de gerar estatísticas e filtrar os resultados a partir de propriedades dos nós.

O projeto Semantic Web Import, portanto, se assemelha mais aos objetivos do presente trabalho, em relação ao Living Semantic Web, porém, com a limitação de estar vinculado à ferramenta GEPHI, ou seja, não é possível analisar as redes geradas com outras ferramentas. Outra limitação está em aceitar apenas consultas SPARQL do tipo CONSTRUCT, excluindo-se a possibilidade de aproveitar consultas dos demais tipos, como SELECT e ASK.

Além das limitações citadas, O Semantic Web Import apresenta inconsistência nos resultados quando comparados aos resultados da mesma consulta executada contra o *endpoint* Virtuoso SPARQL. No teste realizado com a *query* que será apresentada na subseção 4.3.4, adaptada para o tipo CONSTRUCT, o Semantic Web Import trouxe 429 nós e 428 arestas, enquanto o *endpoint* Virtuoso SPARQL apresentou 458 nós e 538 arestas. A Figura 4.3 mostra a janela do *plugin* Semantic Web Import (que foi instalado na ferramenta Gephi versão 0.8.2), onde está configurada a consulta utilizada no teste.

Após o apanhado apresentado nesta subseção do trabalho, uma síntese das principais características dos modelos estudados é apresentada na Tabela 4.5, para fins de comparação. O modelo do presente trabalho está representado pela ferramenta RDFree - resultado do experimento de validação do modelo, que será explanado na próxima seção. As duas primeiras colunas apresentam o nome e propósito do modelo, as terceira e quarta colunas apontam os tipos de dado de entrada e saída dos modelos. Considera-se que a ferramenta RDFree é apenas uma implementação do modelo que, por ser *open-source*,

[3] http://wiki.gephi.org/index.php/SemanticWebImport

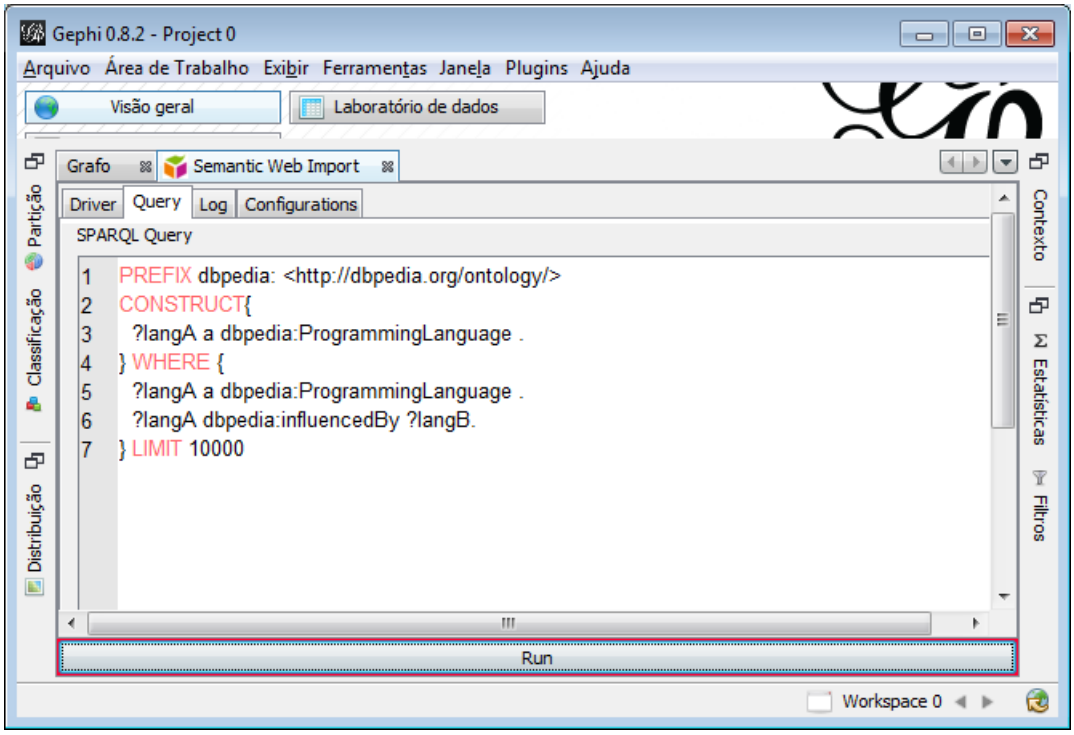

Figura 4.3: Janela do Semantic Web Import, evidenciando a consulta utilizada como teste.

permite a implementaão de vários outros formatos de entrada e saída de dados. A última coluna diz respeito à consistência dos dados, em relação a uma consulta executada contra uma ferramenta de acesso oficial à base de dados DBpedia. A linha correspondente ao modelo Living Semantic Web está em branco porque não utilizou uma *query* para obter os resultados, mas sim, ontologias.

Tabela 4.5: Síntese comparativa entre os modelos.

Modelo	**Propósito**	**Open source**	**Entrada de dados**	**Conversão de dados**	**Consist. dos resultados**
Living Semantic Web	propor um modelo do comportamento da Web Semântica baseado em propriedades de sistemas complexos	não	apenas ontologias	apenas Pajek	-

Continua na próxima página

Tabela 4.5 – Continuação

Modelo	Propósito	Open source	Entrada de dados	Conversão de dados	Consist. dos resultados
Semantic Web Import	importar dados da Web Semântica e estudá-los a partir das propriedades de redes sociais e complexas	não	apenas consultas SPARQL	apenas GEPHI	não
RDFree	propor um modelo para extrair conjutos de dados da Web Semântica e estudá-los a partir das propriedades de redes sociais e complexas	sim	qualquer formato	qualquer formato	sim

4.3 Análise experimental - ferramenta RDFree

Na seções anteriores, a arquitetura genérica e os requisitos do modelo proposto foram descritos e, em seguida, comparados com outros projetos relacionados. O modelo foi desenvolvido de forma a permitir a implementação de diferentes instâncias, a depender da necessidade de cada pesquisador. Como forma de validar esta afirmação, propõe-se um experimento, onde se implementa uma instância de cada um dos módulos do modelo. À ferramenta resultante deste experimento foi dado o nome de RDFree. Trata-se de uma implementação *open-source* hospedada no repositório GitHub, sob o endereço: https://github.com/shankarcabus/RDFree.

Para organizar a leitura, esta seção foi dividida em cinco subseções. Na primeira, é apresentada a delimitação do escopo do experimento e a implementação da ferramenta RDFree. Na segunda subseção, comenta-se os formatos utilizados na implementação da ferramenta: JSON e GEXF. Na terceira parte, o cenário do experimento é apontado e discutido. Já na quarta subseção, este cenário é aplicado à ferramenta e na última são apresentados os resultados obtidos nesta validação e uma discussão acerca deles.

4.3.1 *Escopo do experimento e implementação*

Para viabilizar o experimento, foi necessário delimitar o escopo da ferramenta RDFree, restringindo os requisitos funcionais à implementação de somente um protocolo de coleta e um protocolo de conversão. A Tabela 4.6 apresenta o escopo delimitado para este experimento.

Tabela 4.6: Requisitos funcionais do experimento.

ID	Descrição
RF1	O modelo deve importar dados semiestruturados de uma base de dados Open Linked Data remota, cuja tecnologia é o RDF. O protocolo de coleta utilizará consultas em linguagem SPARQL.
RF2	Os dados devem ser preparados e normalizados para o formato padrão JSON.
RF3	Os dados normalizados devem ser convertidos para o formato GEXF e salvos em um arquivo, que será interpretado pela ferramenta Gephi.

A ferramenta RDFree foi construída a partir dessa delimitação de escopo. A linguagem de programação escolhida foi o Python, por dar suporte às bibliotecas de manipulação de consultas SPARQL e ao formato JSON. Os módulos da arquitetura apresentada na Figura 4.2 podem ser facilmente identificados no código-fonte da ferramenta, pois cada um deles, além do mecanismo controlador, está escrito em arquivos-fonte separados. Cada arquivo-fonte é descrito a seguir:

Main.py – é o arquivo que inicia o programa e controla o fluxo de execução dos outros módulos. Está representado pelo desenho de uma engrenagem, na Figura 4.2.

Extractor.py – é uma implementação do módulo Coletor, responsável pela extração dos dados de uma base de dados RDF. Neste experimento, o protocolo se conecta a bases *Open Linked Data* remotas, e utiliza-se de uma consulta SPARQL para extrair os elementos e seus relacionamentos.

Normalizer.py – é uma implementação do nódulo Normalizador, que recebe os dados extraídos e os transforma para o formato padrão JSON.

Converter.py – é uma implementação do módulo Conversor, que faz a conversão final dos dados normalizados em formato JSON para o formato GEXF.

O RDFree funciona executando-se o arquivo "main.py", que, inicialmente, chama o módulo Extractor. O Extractor busca a consulta SPARQL pré-definida e o endereço do *endpoint*

no arquivo de configuração "config.py". O formato de uma *query* SPARQL aceita pela ferramenta RDFree é semelhante ao apresentado na subseção 3.2.3, com a restrição de que deve retornar exatamente duas variáveis e estas devem ser do mesmo tipo de dado. Essa restrição é representada na cláusula SELECT da *query* e é necessária, pois é a representação de dois sujeitos que se relacionam através dos predicados descritos na cláusula WHERE, os quais serão substituídos pelos valores das n-triplas no retorno da consulta.

Então, o módulo Extractor utiliza uma biblioteca Python chamada "SPARQL Wrapper" para executar a consulta a partir do *endpoint* configurado. À medida que os resultados são retornados, o *prompt* de comando exibe os elementos (nome e id) e os pares relacionados por arestas na tela.

Ao finalizar a consulta, a engrenagem controladora do fluxo chama o módulo Normalizer, passando como parâmetro a estrutura de dados contendo os elementos coletados. O Normalizer separa os nós das arestas e os converte para o formato JSON.

Em seguida, o módulo Converter utiliza a biblioteca "Gexf" para converter os dados normalizados em um formato legível pela ferramenta Gephi e os exporta para o arquivo "output.gexf". Ao final, o programa exibe uma mensagem no tela do prompt de comando, informando o número de nós e arestas encontrados.

O resultado final da execução é o arquivo "output.gexf", o qual o pesquisador deve abrir com a ferramenta Gephi, para visualizar a rede gerada a partir da consulta SPARQL submetida contra a base de dados escolhida. A partir de então, o pesquisador pode fazer análises utilizando os recursos da ferramenta e publicar os resultados.

Nesta subseção, foram citados os dois principais formatos de dados utilizados na implementação do experimento: JSON e GEXF. Para melhor entendimento, eles serão explorados na subseção seguinte.

4.3.2 Delimitação dos formatos GEXF e JSON

As redes complexas são primariamente grafos, portanto, possuem elementos base bem definidos. São eles: arestas e vértices, além de outras informações que podem vir associadas a esses elementos. Partindo-se desse princípio, torna-se trivial pensar em padronizar formatos para o compartilhamento de topologias e dados de redes complexas.

O GEXF surgiu da necessidade de intercambiar informações sobre grafos de forma simples e é o formato padrão da ferramenta Gephi. Neste experimento, o formato JSON está presente no módulo Normalizador, atuando como formato padrão de dados que servirão de

entrada para o módulo Conversor. Já o formato GEXF é utilizado pelo módulo Conversor para gerar um arquivo compatível com a ferramenta de análise de redes Gephi.

As próximas subseções apresentam a ferramenta RDFree inserida em um contexto retirado da Web Semântica, descrevem o processo de construção das *queries* de extração de dados em SPARQL e discute os resultados da análise da rede obtida ao final do processo.

4.3.3 *Cenário*

Para demonstrar a viabilidade da implementação do modelo, a ferramenta RDFree foi submetida a um cenário extraído da base de dados *Open Linked Data*, denominado: **relacionamento de influência entre linguagens de programação no domínio da DBpedia**. O cenário foi escolhido pensando nas inferências que um pesquisador especialista em linguagens de programação pode fazer ao analisar uma rede conectada pela influência que as linguagens exercem umas sobre as outras. A base de dados DBpedia[4] foi escolhida por ser, atualmente, o nó mais conectado no diagrama do projeto *Open Linked Data* (ver Figura 4.4).

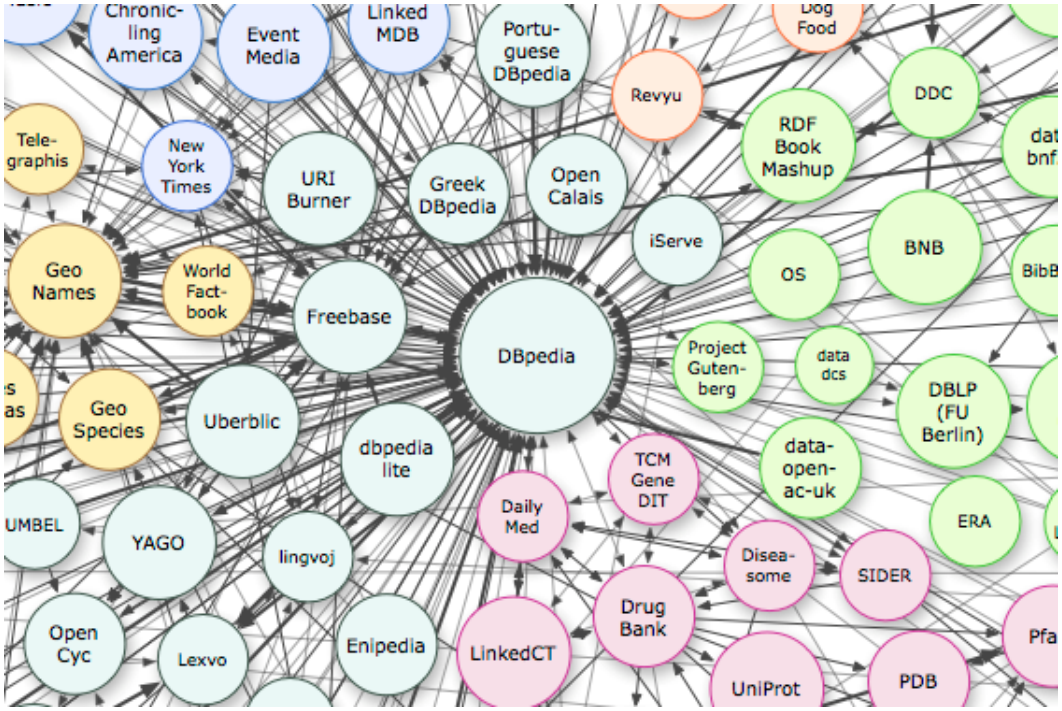

Figura 4.4: Recorte do diagrama *Linking Open Data* evidenciando o *dataset* DBpedia. Fonte: http://lod-cloud.net/versions/2011-09-19/lod-cloud_colored.html.

O objetivo do projeto DBpedia é tornar as informações da Wikipedia disponíveis na Web de forma semiestruturada, transformando, assim, o conteúdo das páginas em mo-

[4] www.dbpedia.org

delos de criação, organização, gestão e difusão do conhecimento de propósitos gerais. Por ser mapeado em RDF, o modelo DBpedia pode tanto ser explorado através de consultas em SPARQL, como também apoiar a implementação de ferramentas da Ciência da Informação, como motores de busca semântica e aplicações que utilizam recursos de inferência.

A base DBpedia foi construída a partir de metadados existentes nas páginas da Wikipedia, mais especificamente, das *infoboxes* - resumos sobre os verbetes da Wikipedia, com objetivo de imprimir consistência à apresentação de artigos de mesma natureza dentro da Wikipedia. Isso porque as informações são organizadas a partir de *templates*[5] de pares "atributo-valor", o que agrega certa estrutura à informação, pois artigos do mesmo tipo utilizam o mesmo *template* de *infobox*. A Figura 4.5 compara um componente da *infobox* sobre a linguagem de programação Python[6] com o mesmo componente extraído da página sobre a linguagem de programação Java[7].

A partir dos pares "atributo-valor" extraídos das *infoboxes*, inicialmente de forma manual e, posteriormente, com auxílio de métodos automatizados de mapeamento, foi criada uma ontologia[8]. No período da realização desta pesquisa, a ontologia da DBpedia estava em sua versão 3.9 com 529 classes, 2.333 propriedades e 3.220.000 instâncias mapeadas (DBPEDIA, 2013). No processo de mapeamento da ontologia, cada assunto, i.e. nome de artigo da Wikipedia, foi convertido para uma classe ou, comparativamente, um sujeito da tripla RDF. Já os atributos das *infoboxes* representam as características ou propriedades desse sujeito, ou seja, os atributos das classes. As relações semânticas são estabelecidas pelo nome do atributo, que se torna o predicado, enquanto o valor do atributo completa a tripla RDF, representando o objeto.

Por exemplo, no excerto de código da *infobox* do artigo sobre "Python (programming language)", representado na Figura 4.6, há os seguintes pares de atributo-valor: "name", "logo", "paradigm", "year", "designer" e "developer".

Na primeira linha, tem-se a indicação do *template* utilizado nessa *infobox*, no caso, "programming language"[9] , que define, entre outras coisas, a classe para onde serão mapeados todos os parâmetros da *infobox*, no caso, a classe "Python" da ontologia. Nas demais linhas estão os pares de atributo-valor, ou predicado-objeto. Para que o parser da Wikipedia funcione, é preciso declarar estritamente os nomes de atributos definidos no *template*. Os valores declarados entre colchetes são transformados pelo *parser* em *links* para

[5] Informações encontradas no artigo sobre *infobox* da própria Wikipedia: http://en.wikipedia.org/wiki/Infobox#Wikipedia

[6] http://en.wikipedia.org/wiki/Python_(programming_language)

[7] http://en.wikipedia.org/wiki/Java_(programming_language)

[8] http://wiki.dbpedia.org/Ontology

[9] A documentação completa do template está disponível em: http://en.wikipedia.org/wiki/Template:Infobox_programming_language.

Figura 4.5: Comparação entre excertos das *infoboxes* dos verbetes "Python (programming language)" e "Java (programming language)". Fonte: http://en.wikipedia.org/.

```
{{Infobox programming language
| name                   = Python
| logo                   = [[Image:Python logo.svg|frameless]]
| paradigm               = [[multi-paradigm programming language|Multi-paradigm]]:
[[object-oriented programming|object-oriented]], [[imperative
programming|imperative]], [[functional programming|functional]], [[procedural
programming|procedural]], [[reflective programming|reflective]]
| year                   = {{start date and age|1991}}
| designer               = [[Guido van Rossum]]
| developer              = [[Python Software Foundation]]
(...)
}}
```

Figura 4.6: Excerto do código da *infobox* do verbete "Python (programming language)". Fonte: http://en.wikipedia.org/w/index.php?title=Python_(programming_language)&action=edit.

os respectivos artigos na Wikipedia. No exemplo acima, pode-se identificar triplas contendo atributos com tipos de dados definidos, como na sentença: ("Python", "paradigm", "multi-paradigm programming language — Multi-paradigm: object-oriented programming — object-oriented"), que significa que Python é uma linguagem multi-paradigma orientada a objetos. Há também triplas formadas por objetos que são textos simples, como em: ("Python", "designer", "Guido van Rossum"), que significa que Python foi criada por Guido van Rossum.

De acordo com DBPedia (2013), o *framework* de mapeamento automatizado foi introduzido somente a partir da versão 3.2 do projeto, o que proporcionou o crescimento mais acelerado da ontologia. Outro benefício conseguido com a automatização foi maior qualidade da informação, pois é possível corrigir as deficiências do sistema de *infoboxes* da Wikipedia, antes de inseri-las na ontologia, e.g. diferentes *infoboxes* para a mesma classe, diferentes nomes para a mesma propriedade e a falta de definição clara de tipos de dados para os valores das propriedades. Atualmente, os mapeamentos usados pelo *framework* estão descritos no documento "DBpedia Mappings Wiki" , onde é possível encontrar as regras usadas na extração, homogeneização e mapeamento entre os *templates* de *infoboxes* a ontologia DBpedia.

Este documento foi largamente utilizado durante a definição do cenário de validação desta pesquisa, denominado: **relacionamento de influência entre linguagens de programação no domínio da DBpedia**. O trabalho iniciou-se com o estudo dos elementos envolvidos com a classe "Programming Language", pois possui o relacionamento "type" com a classe "Python", significando que Python é um tipo de linguagem de programação, classe filha de "Software", que por sua vez é filha de "Work", que é filha de "Thing": a classe mãe de todas as outras em uma ontologia. O mapeamento "Infobox programming language" define para quais classes e tipos de dados os parâmetros serão mapeados. Na Figura 24, mostramos o mapeamento da *infobox* da Figura 4.7.

Outro elemento facilitador para pesquisadores que desejam consultar a base DBpedia é a ferramenta de navegação de dados em RDF: *Open Link Data Explorer*[10], que o projeto DBpedia utiliza para exibir as propriedades de uma classe em formato HTML, quando sua URI é acessada através do browser. A Figura 4.8 mostra parte da classe Python quando acessada através do navegador.

4.3.4 Aplicação da ferramenta

Para montar o experimento, seguimos o objetivo de analisar a influência que as linguagens de programação exercem entre si. A escolha das propriedades que serão usadas no proto-

[10] http://linkeddata.uriburner.com/ode/?uri=http://dbpedia.org/ontology/ProgrammingLanguage

Mapping en:Infobox programming language

Template Mapping (help)	
map to class	ProgrammingLanguage

Mappings

Property Mapping (help)	
template property	name
ontology property	foaf:name

Property Mapping (help)	
template property	designer
ontology property	designer

Property Mapping (help)	
template property	latest_release_version
ontology property	latestReleaseVersion

Property Mapping (help)	
template property	influenced by
ontology property	influencedBy

Property Mapping (help)	
template property	influenced
ontology property	influenced

Figura 4.7: Parte do mapeamento da classe “programming language”. Fonte: http://mappings.dbpedia.org/index.php/Mapping_en:Infobox_programming_language.

About: Python

An Entity of Type : software, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

DBpedia

Property	Value
dbpedia-owl:designer	dbpedia:Guido_van_Rossum
dbpedia-owl:developer	dbpedia:Python_Software_Foundation
dbpedia-owl:influenced	dbpedia:D_(programming_language) dbpedia:JavaScript dbpedia:Falcon_(programming_language) dbpedia:Ruby_(programming_language) dbpedia:Boo_(programming_language) dbpedia:F_Sharp_(programming_language) dbpedia:Groovy_(programming_language) dbpedia:Cobra_(programming_language)
dbpedia-owl:influencedBy	dbpedia:ABC_(programming_language) dbpedia:Dylan_(programming_language) dbpedia:Icon_(programming_language) dbpedia:C++ dbpedia:C_(programming_language) dbpedia:Haskell_(programming_language) dbpedia:Perl dbpedia:ALGOL_68 dbpedia:Modula-3 dbpedia:Java_(programming_language) dbpedia:Lisp_(programming_language)
dbpedia-owl:latestReleaseVersion	2.7.3 / 3.3.0 /
dbpprop:designer	dbpedia:Guido_van_Rossum
dbpprop:developer	dbpedia:Python_Software_Foundation

Figura 4.8: Excerto da classe "Python" visualizada pela ferramenta *OpenLink Data Explorer*. Fonte: `http://live.dbpedia.org/ontology/ProgrammingLanguage`.

colo de extração dos dados pode definir a topologia da rede final. Nas imagens mostradas na subseção anterior, nota-se que existem os atributos "influenced" e "influencedBy" para representar influência. Nesse ponto, pesquisadores que utilizam a base DBpedia podem encontrar dificuldades a respeito da semântica dos elementos envolvidos, pois o significado semântico não está documentado claramente no projeto. Por isso, além de consultar os recursos discutidos até aqui - *templates* da Wikipedia e mapeamentos do projeto DBpedia, é necessário executar *queries* com diferentes combinações de classes e propriedades, até encontrar aquelas mais apropriadas. Para isso, foi utilizado neste trabalho o *endpoint* Virtuoso SPARQL, mencionado anteriormente na seção 3.2.3.

Comparativamente, a *query* que utiliza a propriedade "influencedBy" retorna uma lista de 990 registros (que serão convertidos em vértices da rede), os quais relacionam as linguagens de programação àquelas que as influenciam. Já se optarmos pela propriedade "influenced", a *query* retorna 538 registros, pois agora são listadas as linguagens de programação relacionadas àquelas que ela influencia. Optamos por utilizar a segunda opção de *query*, que está representada na Figura 4.9, pois permite identificar as linguagens de programação que mais influenciaram outras linguagens e destacá-las na rede final. Essa *query* traz os nomes de todas as linguagens de programação disponíveis na base DBpedia ("langA") relacionadas às linguagens de programação que elas influenciam ("langB").

```
1  PREFIX dbpedia: <http://dbpedia.org/ontology/>
2  SELECT ?langA, ?langB
3  WHERE {
4      ?langA a dbpedia:ProgrammingLanguage .
5      ?langA dbpedia:influencedBy ?langB.
6  }
```

Figura 4.9: *Query* utilizando a propriedade "influencedBy‘" para retornar o domínio dos relacionamentos entre linguagens de programação.

Para configurar o protocolo de extração deste cenário, tanto o endereço do *endpoint* como a *query* escolhida devem ser adicionados ao arquivo de configuração da ferramenta RDFree, conforme indicado na Figura 4.10.

Após configurar a ferramenta, aciona-se o arquivo "main.py", para dar início à sequência de conversões executadas pelos módulos, as quais foram explicadas detalhadamente na subseção 4.3.1 deste trabalho. O módulo Coletor utiliza as informações de configuração para extrair uma coleção de nós e arestas a partir do domínio selecionado. Já no módulo normalizador, esta coleção é convertida para o formato JSON (ver subseção 4.3.2), que consiste em uma lista de pares de elementos (linguagens de programação), que são as variáveis "langA" e "langB" e seus valores. Para cada relacionamento de influência encontrado, é criado um par nesta listagem. A Figura 4.11 mostra uma parte do arquivo JSON gerado pelo módulo normalizador, onde percebe-se algumas das linguagens que

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

OUTPUT_FILE = 'output.gexf'
ENDPOINT = "http://dbpedia.org/sparql"
QUERY = '''
PREFIX dbpedia: <http://dbpedia.org/ontology/>
SELECT ?langA, ?langB
WHERE {
    ?langA a dbpedia:ProgrammingLanguage .
    ?langA dbpedia:influencedBy ?langB.
}
'''

# GEXF
CREATOR = 'Shankar Cabus de Teive e Argollo'
DESCRIPTION = u'Aplicando a solucao em um dominio'
DEFAULTEDGETYPE = 'undirected'
MODE = 'static'
LABEL = 'Relacionamento entre linguagens de programacao'
```

Figura 4.10: Arquivo “config.py” configurado para o domínio dos relacionamentos entre linguagens de programação.

influenciaram a linguagem Python.

```
{ "langA": { "type": "uri", "value": "http://dbpedia.org/resource/Python_(programming_language)" } ,
"langB": { "type": "uri", "value": "http://dbpedia.org/resource/Haskell_(programming_language)" }},
{ "langA": { "type": "uri", "value": "http://dbpedia.org/resource/Python_(programming_language)" } ,
"langB": { "type": "uri", "value": "http://dbpedia.org/resource/C_(programming_language)" }},
{ "langA": { "type": "uri", "value": "http://dbpedia.org/resource/Python_(programming_language)" } ,
"langB": { "type": "uri", "value": "http://dbpedia.org/resource/Dylan_(programming_language)" }},
{ "langA": { "type": "uri", "value": "http://dbpedia.org/resource/Python_(programming_language)" } ,
"langB": { "type": "uri", "value": "http://dbpedia.org/resource/Icon_(programming_language)" }},
{ "langA": { "type": "uri", "value": "http://dbpedia.org/resource/Python_(programming_language)" } ,
"langB": { "type": "uri", "value": "http://dbpedia.org/resource/Java_(programming_language)" }},
{ "langA": { "type": "uri", "value": "http://dbpedia.org/resource/Python_(programming_language)" } ,
"langB": { "type": "uri", "value": "http://dbpedia.org/resource/Lisp_(programming_language)" }}
```

Figura 4.11: Excerto do arquivo JSON gerado pelo módulo Normalizador.

O módulo Conversor varre o arquivo JSON para extrair os pares de elementos e traduzi-los para o formato compatível com a ferramenta Gephi: o GEXF. Este formato é baseado em XML, ou seja, está organizado com *tags*, de forma hierárquica, conforme pode-se observar na Figura 4.12. A *tag* raiz é a “gexf”, onde se aninham as tags “meta” - que abriga os metadados do arquivo - e “graph”. Dentro de “graph” existem as *tags* “nodes”, onde cada nó da rede está aninhado, e “edges”, que relaciona os pares de nós da rede. Cada nó possui os atributos “id” e “label” (identificador númerico e rótulo do nó, respectivamente). Já as arestas, ou *edges*, são compostas dos atributos “id”, “source” e “target”, que são o identificador numérico da aresta, o identificador do nó de partida e o identificador do nó de chegada do arco, respectivamente.

```
<gexf xmlns:viz="http://www.gexf.net/1.1draft/viz" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.gephi.org/gexf/1.1draft" xmlns:ns0="xsi" version="1.1" ns0:schemaLocation="
http://www.gephi.org/gexf/1.1draft http://gephi.org/gexf/1.1draft.xsd">
  <meta lastmodified="2014-09-29T00:52:05.003000">
    <creator>Shankar Cabus de Teive e Argollo</creator>
    <description>Aplicando a solucao em um dominio</description>
  </meta>
  <graph defaultedgetype="undirected" label="Relacionamento entre linguagens de programacao" mode="static">
    <nodes>
      <node id="1" label="http://dbpedia.org/resource/Boo_(programming_language)">
        <attvalues/>
      </node>
      <node id="2" label="http://dbpedia.org/resource/Python_(programming_language)">
        <attvalues/>
      </node>
      <node id="3" label="http://dbpedia.org/resource/C_Sharp_(programming_language)">
        <attvalues/>
      </node>
      ...
    </nodes>
    <edges>
      <edge id="0" source="1" target="2">
        <attvalues/>
      </edge>
      <edge id="1" source="1" target="3">
        <attvalues/>
      </edge>
      ...
    </edges>
  </graph>
</gexf>
```

Figura 4.12: Excerto do arquivo GEXF gerado pelo módulo Conversor.

Esse arquivo é salvo com o nome de "output.gexf" na pasta raiz. Para visualizar a rede, é preciso abri-lo com a ferramenta Gephi. Na próxima subseção, a rede gerada será explorada através dessa ferramenta e alguns resultados serão discutidos.

4.3.5 Resultados e discussão

A forma de apresentação dos dados é muito importante para um pesquisador interessado em analisar e tirar conclusões sobre um determinado conjunto de informações. Este trabalho reuniu dados abertos encontrados na Web Semântica e os converteu em um modelo de representação do conhecimento: uma rede complexa. Dessa forma, o pesquisador não tem apenas dados, mas sim, um modelo onde as informações estão conectadas, gerando conhecimento.

Ao abrir o arquivo "output.gexf" gerado pela ferramenta RDFree com a ferramenta Gephi, os pesquisadores podem visualizar a rede e fazer análises topológicas, através de cálculos dos parâmetros de redes sociais e complexas. É possível também alterar o *layout* de exibição da rede de acordo com os valores calculados e também publicar os resultados em diferentes formatos. A Figura 4.13 expõe a rede gerada após executar alguns algoritmos de *layout* e ajustar a cor e o tamanho dos vértices de acordo com seu grau.

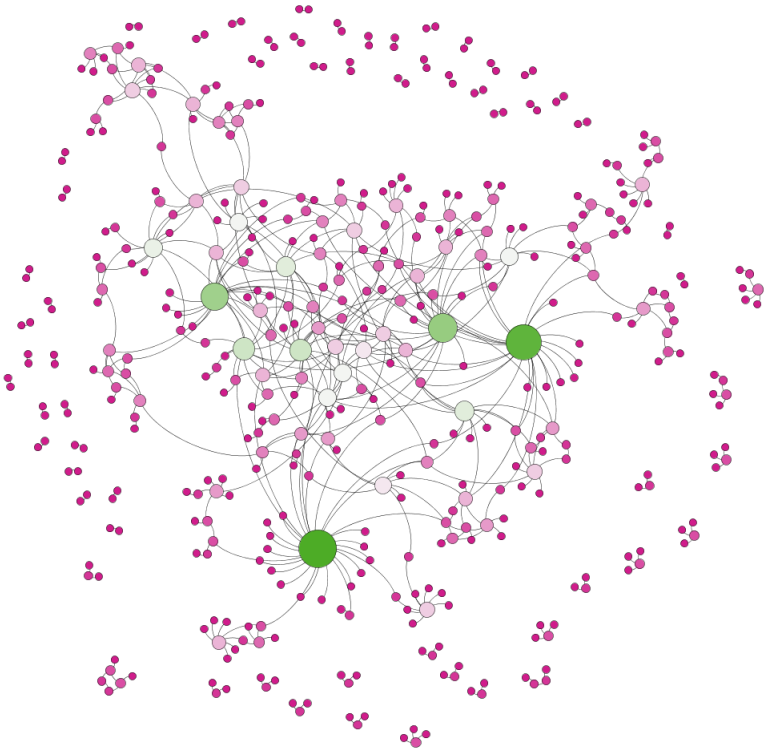

Figura 4.13: Rede do relacionamento de influência entre linguagens de programação.

A rede gerada possui 458 vértices, 538 arestas e 63 componentes. O componente gigante da rede possui 300 vértices e 441 arestas, o que representa mais de 60% da rede completa. A Figura 4.14 apresenta a rede formada pelo componente gigante.

Neste caso, as duas redes serão analisadas paralelamente, nomeadas de: a) rede completa e; b) componente gigante. Para iniciar a análise, foram calculadas as seguintes propriedades, representadas na Tabela 5: número de vértices (n), número de arestas (m), grau médio ($< k >$), densidade (Δ), coeficiente de aglomeração médio ($< C >$), caminho mínimo médio ($< l >$), diâmetro (D) e quantidade de componentes.

Tabela 4.7: Quadro de propriedades da rede.

Propriedade	**Rede Completa**	**Componente Gigante**
$n = \|V\|$	458	300 (65,5%)
$m = \|E\|$	538	441(81,97%)
$< k >$	2,349	2,94
Δ	0,005	0,01
$< l >$	5,245	5,258
D	12	12

Continua na próxima página

Tabela 4.7 – Continuação

Propriedade	Rede Completa	Componente Gigante
Componentes	63	1

A propriedade $< k >$ indica que cada nó está conectado, em média, a 2,349 outros nós, ou seja, a conectividade média da rede é baixa. Ao analisar apenas o componente gigante, o grau médio aumenta para 2,94, pois foram eliminados os nós isolados, que possuem poucas conexões. A propriedade $< C >$ não foi mostrada na tabela porque seu valor é igual à zero, o que indica falta de aglomeração na rede, ou seja, há poucas conexões entre os nós adjacentes a um determinado vértice. Isso pode indicar que duas linguagens influenciadas pela mesma linguagem de programação não influenciam umas às outras, por sua vez, ou que essa influência é insignificante. A propriedade $< l >$ indica que o menor caminho médio entre dois nós da rede é 5,245.

Apesar do coeficiente de aglomeração nulo, pode-se observar que há vértices que se conectam a outros que, por sua vez, são também muito conectados. É o caso da linguagem de programação C, que embora não tenha o maior grau, exerce um grande poder de influência, a despeito das linguagens com maior quantidade de conexões: LISP, Haskell e Smalltalk. Isso porque a linguagem C está conectada a outras linguagens influentes, como Java, C++, Python, dentre outras, como mostra a Figura 4.15.

A distribuição de graus da rede completa é regida por uma lei de potência, com γ=-1,8 (aproximadamente), o que pode indicar uma topologia livre de escala. A Figura 4.16 apresenta o gráfico da função de distribuição de graus ajustada para log-log.

Apesar de sua natureza exata, os valores das propriedades de redes apresentadas ao longo deste trabalho podem ser investigadas subjetivamente, sob diversos ângulos, a depender do objetivo de pesquisa de cada especialista. Assim, surge a questão: mesmo com todo o arcabouço que a disciplina de redes complexas disponibiliza, os gráficos gerados mostram as mesmas propriedades estatísticas e comportamento que as redes reais que eles modelam? Como a disciplina de redes complexas é relativamente nova, essa questão vem sendo refinada a cada trabalho publicado. Neles, as propriedades são especialmente importantes para a modelagem de redes de grande escala, característica que impossibilitaria até mesmo sua análise por outros meios. Com base nessa reflexão, o capítulo seguinte apresentará as conclusões obtidas pelo presente trabalho.

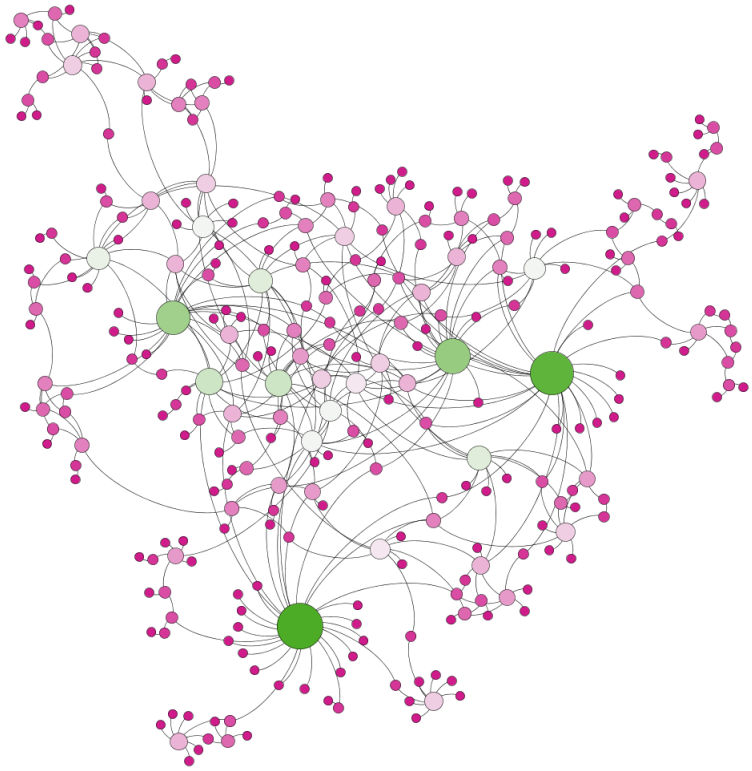

Figura 4.14: Componente gigante da rede.

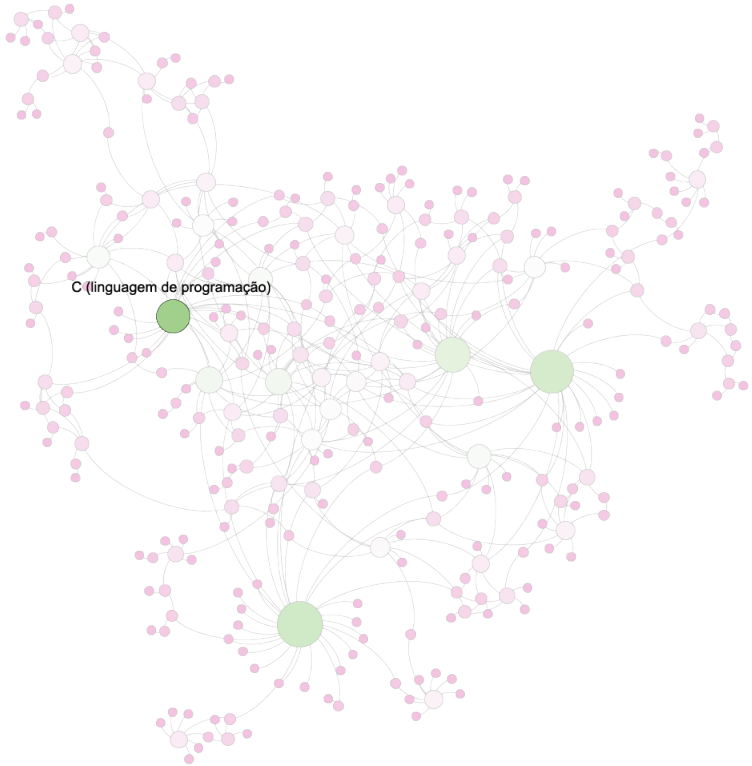

Figura 4.15: Componente gigante da rede, com evidência para a linguagem C.

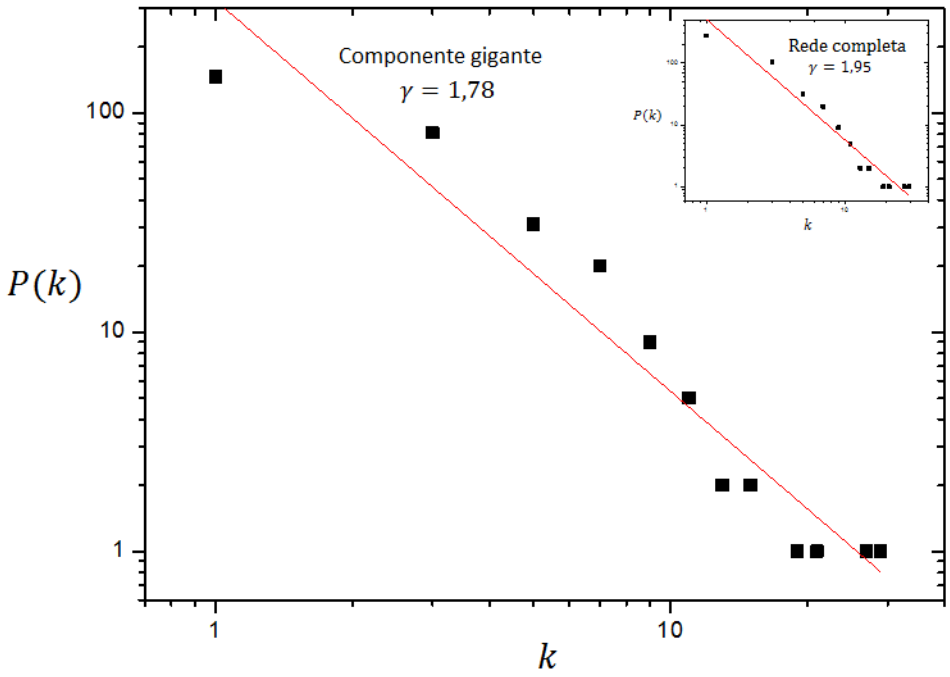

Figura 4.16: Distribuição de graus do componente gigante ajustada para log-log. O gráfico menor apresenta a distribuição de graus da rede completa, para comparação.

Capítulo Cinco

Considerações finais

Este capítulo apresenta o apanhado geral sobre o trabalho de pesquisa realizado e os resultados obtidos.

5.1 Conclusões

O presente trabalho de pesquisa prezou por explorar a Web Semântica através de formatos que pudessem contribuir para o trabalho de outros pesquisadores, de diversas áreas do conhecimento. Ao analisar um conjunto de dados sob a perspectiva da teoria de redes, pesquisadores poderiam tirar proveito de propriedades e inferências que esse modelo proporciona.

Para realização desta pesquisa, seguiu-se uma metodologia onde os objetivos específicos foram alcançados gradativamente, em cada etapa. Primeiramente, foi feito o levantamento bibliográfico sobre os conteúdos envolvidos. Em seguida, o modelo genérico foi especificado, por meio da descrição da sua arquitetura, bem como de seus requisitos funcionais e não funcionais. Depois, foram feitas restrições ao modelo, para viabilizar a construção da ferramenta RDFree, como forma de validação do modelo genérico. Essa etapa subdividiu-se em duas: a primeira caracterizou-se pela implementação da ferramenta e a segunda, pela definição de um protocolo de extração de dados de repositórios RDF e outro de conversão dos dados para o formato GEXF.

Posteriormente, a ferramenta foi submetida a um experimento envolvendo a base de dados DBpedia, no domínio do relacionamento de influência entre linguagens de programação. Como resultado do experimento, foi gerada uma rede de linguagens de programação conectadas entre si pelo relacionamento de influência. Por fim, as propriedades da rede complexa foram calculadas e suas relações semânticas foram mapeadas e discutidas, o que gerou algumas inferências, como a descoberta de linguagens de programação que encurtam o caminho mínimo médio da rede, pois influenciam direta e indiretamente uma grande parte dos nós (Ver subseção 4.3.5 deste trabalho).

Assim, conclui-se que os objetivos propostos foram alcançados, dentro dos limites estabelecidos, visto que o pesquisador obteve êxito em sua pesquisa, ao utilizar a ferramenta RDFree para converter os dados da Web Semântica em um modelo de redes sociais e complexas. Embora este trabalho consista em propor um modelo computacional flexível, que

aceita a vinculação de diferentes protocolos de extração e conversão dos dados, os limites estabelecem que a validação do modelo seria feita a partir de uma delimitação de um protocolo de coleta - RDF e SPARQL - e um protocolo de conversão - ferramenta GEPHI, conforme seção 1.4. A partir dessa delimitação foi construída a ferramenta RDFree e esta foi validada com a aplicação de um cenário experimental: linguagens de programação no domínio DBpedia. A evidência disso foi apresentada na subseção 4.3.5, onde os resultados da análise da rede foram discutidos.

O modelo é também considerado consistente, pois, ao ser executado seguidas vezes, apresenta sempre o mesmo resultado. Esta conclusão foi obtida avaliando-se quantitativamente os nós e arestas obtidos a partir de uma mesma consulta SPARQL. O modelo também obteve êxito ao ser comparado com os resultados obtidos pelo *endpoint* Virtuoso SPARQL, ao contrário do modelo Semantic Web Import, conforme demonstrado na seção 4.2. Entretanto, para que o modelo seja verificado em contextos mais amplos, é necessário definir novos protocolos de extração e conversão, implementá-los na ferramenta RDFree e submetê-los a novos experimentos de validação.

5.2 Contribuições

Este trabalho contribui de forma teórica para a comunidade acadêmica, pois apresenta uma síntese das áreas de pesquisa e as relaciona, de forma a ampliar o alcance de suas descobertas. Além disso, traz contribuições práticas importantes, pois o modelo proposto permite que pesquisadores tirem maior proveito de dados públicos em suas pesquisas, conforme foi demonstrado pelos resultados alcançados. O aspecto da modularização da arquitetura também pode ser considerado uma contribuição prática, pois amplia o leque de possibilidades de uso do modelo e deixa em aberto questões para pesquisas futuras, algumas das quais serão discutidas na próxima seção.

5.3 Atividades Futuras de Pesquisa

Como trabalhos futuros, sugere-se que o modelo seja enriquecido, com o acréscimo de outros formatos de arquivos de entrada, como, por exemplo, o PDF. Por tratar-se de um formato não estruturado, é necessário realizar um estudo sobre mineração de textos e incluir tratamentos específicos ao protocolo de extração, para garantir o máximo de confiabilidade dos dados. É válido ressaltar que quanto menos estruturados forem os dados de entrada, mais específicas e detalhadas serão as diretrizes do protocolo, como também será mais baixo o desempenho da ferramenta.

Sugere-se também a inclusão de outros protocolos de conversão, para gerar maior variedade de formatos de saída de dados do modelo. Um exemplo é o formato utilizado pela ferramenta Pajek[1] -".net", também usado por outras ferramentas de análise de redes.

[1] http://pajek.imfm.si/doku.php

Referências Bibliográficas

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, American Physical Society, v. 74, n. 1, p. 47–97, 2002. Disponível em: <http://link.aps.org/doi/10.1103/RevModPhys.74.47>.

ALBERT, R.; JEONG, H.; BARABÁSI, A.-L. Internet: Diameter of the world-wide web. *Nature*, Nature Publishing Group, v. 401, n. 6749, p. 130–131, 1999.

ALBERT, R.; JEONG, H.; BARABÁSI, A.-L. Error and attack tolerance of complex networks. *Nature*, v. 406, 2000.

ARGOLLO, S. C. d. T. Uma solução computacional para integração entre web semântica e redes complexas. Monografia (Bacharelado em Sistemas de Informação) - Universidade do Estado da Bahia, Salvador. 2012.

BARABÁSI, A.-L. The architecture of complexity, from network structure to human dynamics. *Ieee Control Systems Magazine*, p. 33–42, 2007.

BARABÁSI, A.-L. *Linked: A Nova Ciência dos Networks*. São Paulo: Leopardo, 2009.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, v. 286, p. 509–512, 1999.

BARABÁSI, A.-L.; ALBERT, R.; JEONG, H. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications*, v. 281, n. 1-4, p. 69–77, jun. 2000.

BERNERS-LEE, T. *Linked Data: Design Issues*. 2001. Disponível em: <http://www-.w3.org/DesignIssues/LinkedData.html>.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, p. 33–42, 2001.

BOCCALETTI, S.; LATORA, V.; MORENO, Y.; CHAVEZ, M.; HWANG, D.-U. Complex networks : Structure and dynamics. *Phys. Rep.*, v. 424, n. 4-5, p. 175–308, 2006.

BOLLOBAS, B. *Modern Graph Theory*. New York: Springer, 1998. (Graduate Texts in Mathematics, v. 184).

BRAY, T. *RDF and Metadata*. 1998. Disponível em: <http://www.xml.com/pub/a/98-/06/rdf.html>.

BUCHANAN, M. *Nexus: fundamentos da ciência dos networks*. São Paulo: Leopardo, 2010.

COHEN, R.; EREZ, K.; BEN-AVRAHAM, D.; HAVLIN, S. Resilience of the internet to random breakdowns. *Physical Review Letters*, p. 4626–4628, 2000.

DBPEDIA. *The DBPedia Ontology (3.9)*. 2013. Disponível em: <http://wiki.dbpedia-.org/Ontology39?v=g9b>.

DIESTEL, R. *Graph Theory*. Heidelberg: Springer-Verlag, 2010. (Graduate Texts in Mathematics, v. 173).

ERDöS, P.; RéNYI, A. On random graphs . *Publicationes Mathematicae Debrecen*, v. 6, p. 290–297, 1959.

ERDöS, P.; RéNYI, A. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, v. 5, p. 17–61, 1960.

EULER, L. Solutio problematis ad geometriam situs pertinentis. *Graph Theory 1736-1936*, Oxford University Press, USA, 1736.

FEIGENBAUM, L.; PRUD'HOMMEAUX, E. *SPARQL by Example*. [S.l.], maio 2013. W3C Recommendation. Disponível em: <http://www.cambridgesemantics.com-/semantic-university/sparql-by-example>.

GIL, R.; GARCÍA, R. Measuring the semantic web. *First on-Line conference on Metadata and Semantics Research, MTSR*, Rinton Press, 2005.

HEATH, T.; BIZER; CHRISTIAN. *Linked Data: Evolving the Web into a Global Data Space*. [S.l.]: Morgan & Claypool, 2011. (Synthesis Lectures on the Semantic Web: Theory and Technology, v. 1).

LIEBSCHER, P. Quantity with quality? teaching quantitative and qualitative methods in an lis master's program. *Library Trends*, Spring, v. 46, n. 4, p. 668–680, 1998.

MILGRAM, S. The small world problem. *Psychology today*, New York, v. 2, n. 1, p. 60–67, 1967.

MOTTER, E. Cascade control and defense in complex networks. *Physical Review Letters*, n. 93, 2004.

NEWMAN, M. E. J. The structure and function of complex networks. *SIAM Review*, v. 45, n. 2, p. 167–256, 2003.

NEWMAN, M. E. J.; BARABÁSI, A.-L.; WATTS, D. J. *The Structure and Dynamics of Networks*. Princeton: Princeton University Press, 2006.

NUSSENSVEIG, H. M. *Complexidade e Caos*. Princeton: UFRJ Editora, 2008.

REDNER, S. How popular is your paper? an empirical study of the citation distribution. *The European Physical Journal B*, p. 131–134, 1998.

SOLOMONOFF R.; RAPOPORT, A. Connectivity of random nets. *The bulletin of mathematical biophysics*, v. 13, n. 2, p. 107–117, 1951.

TAUBERER, J. *What Is RDF*. 2006. Disponível em: <http://www.xml.com/pub/a-/2001/01/24/rdf.html>.

W3C, R. W. G. *RDF 1.1 Primer*. [S.l.], junho 2004. W3C Working Group Note. Disponível em: <http://www.w3.org/TR/rdf11-primer/>.

W3C, S. W. G. *SPARQL 1.1 Overview*. [S.l.], março 2013. W3C Recommendation. Disponível em: <http://www.w3.org/TR/sparql11-overview/>.

W3C, S. W. G. *SPARQL 1.1 Protocol*. [S.l.], março 2013. W3C Recommendation. Disponível em: <http://www.w3.org/TR/sparql11-protocol/>.

W3C, T. A. G. *Architecture of the World Wide Web*. [S.l.], dezembro 2004. W3C Recommendation. Disponível em: <http://www.w3.org/TR/webarch>.

WASSERMAN, S.; FAUST, K. *Social Network Analsis*. [S.l.]: Cambridge: Cambridge University Press., 1994.

WATTS, D. J. *Seis Graus de Separação: a evolução da ciência de redes em uma era conectada*. São Paulo: Leopardo, 2009.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of'small-world'networks. *Nature*, v. 393, n. 6684, p. 409–10, 1998.

Modelo computacional para analisar dados semiestruturados na Web Semântica com o auxílio da teoria de redes

Gabriela Oliveira Mota da Silva

Salvador, Outubro de 2014.