Uma Abordagem Baseada em Ontologias para a Predição de Ligações em Redes de Colaboração Científica

Thiago Henrique Dias Araujo

Texto apresentado
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
o Exame de Qualificação
de
Mestre em Ciências

Programa: Mestrado em Ciência da Computação

Orientadora: Prof[a]. Dr[a]. Renata Wassermann

São Paulo, junho de 2016

Uma Abordagem Baseada em Ontologias para a Predição de Ligações em Redes de Colaboração Científica

Esta é a versão original do texto elaborado pelo candidato Thiago Henrique Dias Araujo para o exame de qualificação apresentado ao Instituto de Matemática e Estatística da Universidade de São Paulo como requisito para obtenção de título de Mestre em Ciências.

Resumo

ARAUJO, T. H. D. **Uma Abordagem Baseada em Ontologias para a Predição de Ligações em Redes de Colaboração Científica**. 2016. 36 f. Exame de qualificação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

A comunidade científica pode ser enxergada como uma rede em que cada pesquisador se relaciona com outros através de colaborações diversas como, por exemplo, na coautoria de um artigo científico. Alguns trabalhos aplicam técnicas de aprendizado de máquina para prever ligações entre os participantes de uma rede, tratando do problema conhecido como Predição de Ligações. Entretanto, algumas dessas metodologias apresentam certas limitações, ou por utilizarem formas pouco expressivas de representação do domínio, ou por analisarem apenas a estrutura da rede, sem levar em consideração as características intrínsecas dos participantes dessa rede. A proposta do presente trabalho é modelar uma ontologia capaz de indicar características próprias dos pesquisadores da Plataforma Lattes e suas relações com outros pesquisadores, extraindo conhecimento prévio sobre o domínio e, posteriormente, utilizá-lo no enriquecimento de um modelo de aprendizado que faça predição de novas colaborações. Esperamos que esta abordagem aqui apresentada seja mais eficaz do que as metodologias exploradas na literatura, e que contribua para uma melhor colaboração entre pesquisadores.

Palavras-chave: redes de colaboração científica, ontologia, aprendizado de máquina, predição de ligações, plataforma Lattes.

Abstract

ARAUJO, T. H. D. **Uma Abordagem Baseada em Ontologias para a Predição de Ligações em Redes de Colaboração Científica**. 2016. 36 f. Exame de qualificação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

The scientific community can be seen as a network of people whose relationships are built by means of various types of collaborations, like the co-authoring of a scientific paper. Some applications of machine learning to the problem of link prediction have been done before, but these methods have some limitations because they employ representations of the domain that are not very expressive. And some of these works only analyse the structure of these networks, without taking into consideration the characteristics and attributes of the people that integrate these networks. Our proposal is to create an ontology able to show the characteristics of researchers and their relationships with others, using information from Plataforma Lattes, and extract background-knowledge about this domain that will be used to enrich a machine learning model capable of predicting new collaborations and co-authorships. We hope that our approach will be more efficient than other existing methodologies, and that it will help researchers form better relationships and collaborations.

Keywords: scientific collaboration networks, ontology, machine learning, link prediction, plataforma Lattes.

Sumário

Lista de Abreviaturas

ILP	(*Inductive Logic Programming*)
ODBA	(*Ontology-Based Data Access*)
OWL	(*Web Ontology Language*)
SGBD	(*Sistema de Gerenciamento de Banco de Dados*)
SPARQL	(*SPARQL Protocol and RDF Query Language*)
SQL	(*Structured Query Language*)
SRL	(*Statistical Relational Learning*)
SVM	(*Support Vector Machine*)
SWRL	(*Semantic Web Rule Language*)

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A internet não é apenas um repositório quase infinito de informações aleatórias em páginas da web. Em uma análise mais apurada, ela passa a ilustrar imensas cadeias de significados. A Extração de Conhecimento (*Knowledge Discovery in Databases*) é um processo que aplica ferramentas e algoritmos de Mineração de Dados na análise desse imenso conjunto de dados, e tem por objetivo a extração de informações e a descoberta e construção de novos conhecimentos úteis (Fayyad *et al.*, 1996).

Por essa razão, quando falamos da quantidade maciça de informação disponível, conhecida como *Big Data*, devemos pensar na grande quantidade de **significados** que podemos extrair desses dados. Para Kay (2014), o que chamamos de *Big Data* não é algo relevante pela sua quantidade, mas sim pelo que ele chama de *Big Meaning*, isto é, toda essa imensa riqueza de significados presente na web que podemos estudar.

Nesse panorama, surgem novas possibilidades de extração de informação e de construção do conhecimento de forma automatizada. Através de modelos de classificação e de análise, tornou-se possível, segundo Halevy *et al.* (2009), construir sistemas que entendem uma frase em determinada língua e a traduzem para outra, ou que interpretam e classificam textos, tudo isso usando como exemplo apenas os inúmeros textos contidos na web.

Ainda como desdobramento dessas novas técnicas, tornou-se possível analisar as interações entre seres humanos em redes sociais, como nas redes de colaboração científica, que contêm informações de autores, publicações e seus temas, em diversas áreas do saber. É possível identificar comunidades interessadas em temas similares, grupos de pesquisadores e mudanças nos interesses dessas comunidades. Através disto, é possível saber mais a respeito do dia-a-dia da produção científica e do desenvolvimento da Ciência em geral ao longo do tempo.

Com especial interesse, podemos analisar essas redes de colaboração, que formam um subconjunto da comunidade científica e avaliar o desenvolvimento da Ciência em todo o mundo, descobrindo quais são as principais contribuições e tendências, como se dá o relacionamento entre pesquisadores e qual a importância e influência de grupos de pesquisa em suas comunidades. Esse tipo de análise é interessante em levantamentos sobre produtividade e impacto de uma pesquisa, relevância e popularidade de um projeto ou uma área, como formas de se mensurar e quantificar o progresso científico, entre outros aspectos.

Várias técnicas de análise podem ser empregadas nesse problema, desde as mais simples, como relatórios de produtividade, até as mais complexas, como o agrupamento de pesquisadores em categorias específicas ou a classificação de trabalhos de forma automática. Na literatura, encontramos algumas ferramentas muito úteis para essa análise. Uma delas trata do problema da predição de ligações entre pesquisadores, que permite prever, com boa segurança, novas colaborações, através da verificação das características de trabalhos publicados, interesses de pesquisa, e outras particularidades.

Como é usual no método científico, encontramos algumas limitações nessas técnicas, geralmente derivadas de métodos da área de Aprendizado de Máquina. Algumas não levam em consideração características dos pesquisadores, mas apenas da estrutura da rede. Existem representações mais

expressivas que poderiam ser aplicadas ao domínio com facilidade, e várias técnicas relacionadas à área de Sistemas Baseados em Conhecimento que possibilitariam simplificar essa tarefa e aumentar sua eficácia.

Com o presente trabalho, esperamos contribuir para esse tipo de análise, aplicando ideias vindas da área de Sistemas Baseados em Conhecimento, e comparando o resultado com outros trabalhos, a fim de analisar a possível eficácia das técnicas propostas.

1.1 Motivação

Uma rede de colaboração científica, como outras redes sociais, é uma estrutura relacional que podemos facilmente representar por um grafo com atributos, como fez Cervantes (2014), ou por uma ontologia, como a desenvolvida por Costa e Yamate (2009), ou até mesmo por um banco de dados relacional. Também possui uma natureza multi-relacional, pois uma publicação pode ter vários tipos de relacionamento diferentes, como citações, autoria, veículo de publicação, dentre outros. E um autor pode se relacionar com outro via coautoria de um artigo, participação na mesma banca ou conferência, fazendo parte do mesmo grupo de pesquisa, ou tendo interesses similares, dentre inúmeras outras relações possíveis. Todos esses atributos podem contribuir na análise da estrutura desse tipo de rede.

Entretanto, se desejamos aplicar um modelo de aprendizado para analisar alguma característica da rede, geralmente utilizamos modelos que recebem como entrada tabelas de atributos e valores. Partindo dessa abordagem, faz-se necessário transformar uma representação relacional mais rica (como um grafo) em uma representação mais simples (como uma tabela), que nos leva a dizer que uma ferramenta de aprendizado supervisionado desse tipo trabalha apenas com representações pouco expressivas. Podemos chamar isso de um "problema de expressividade".

Segundo Raedt (2008), existem várias representações possíveis na modelagem de um problema de aprendizado, algumas mais expressivas, outras menos. Ele mostra em seu trabalho que existe uma hierarquia dessas representações em uma ordem crescente de expressividade. Discutimos essa ideia no capítulo sobre trabalhos correlatos.

Além da análise da estrutura da rede, podemos investigar outras características, como o perfil de um pesquisador, seu currículo, sua experiência e quem são seus pares, ou a área de pesquisa em que estão inseridas as suas publicações mais importantes. Porém, não é nada trivial transformar esse conhecimento em um conjunto de tabelas de atributos e valores.

A principal motivação deste trabalho é a possibilidade de se utilizar conhecimento prévio do domínio, também chamado de *background-knowledge*. A proposta é construir uma ontologia que representa o conhecimento acerca de uma rede de colaboração, sendo capaz de inferir novas relações, regras e derivar novas características das entidades pertencentes ao domínio. Assim, é possível extrair esse conhecimento (*background-knowledge*) e transformá-lo em novos atributos das entidades presentes na rede, enriquecendo os dados a serem explorados pelo modelo de predição.

A principal vantagem do uso de uma ontologia é que o conhecimento extraído é declarativo, compacto e de alto-nível, o que simplifica a sua validação e apreciação. Sem contar que esse conhecimento a respeito de uma entidade do domínio pode se espalhar para outras entidades, gerando outros novos atributos. Esperamos, com isso, melhorar a eficácia do modelo.

1.2 Objetivos

O objetivo do presente trabalho é demonstrar que um modelo de predição de ligações em uma rede de colaboração científica, quando enriquecido com conhecimento prévio (*background knowledge*) extraído de uma ontologia, torna-se mais eficaz em sua tarefa de predição. Para isso, espera-se cumprir os seguintes passos:

- Modelar uma Ontologia para representar o conhecimento a respeito de uma rede de colaboração científica, contendo informações sobre pesquisadores, publicações, colaborações e insti-

tuições.

- Construir consultas em SPARQL na ontologia, as quais serão utilizadas como conhecimento prévio. Algumas dessas consultas são:
  - Qual a área de pesquisa de um pesquisador?
  - Quem são os pesquisadores que ele orientou?
  - Há quantos anos um dado pesquisador trabalha em uma instituição?
- Construir um modelo de predição de ligações baseado no trabalho de Cervantes (2014).
- Enriquecer o modelo de predição com o conhecimento prévio extraído da ontologia, executá-lo, e comparar sua eficácia com o modelo básico.
- Propor novas consultas à ontologia que possam ser utilizadas como conhecimento prévio aplicado ao modelo e fazer novos testes.

1.3 Metodologia

Os objetivos traçados serão alcançados através da execução dos seguintes itens:

1. **Estudo dos resultados de Cervantes (2014)**:
   O objetivo desse item é entender o modelo de representação das redes de colaboração como grafos com atributos e o modelo de predição utilizando aprendizado supervisionado para que possa ser implementado e enriquecido.
2. **Estudo sobre modelagem de Ontologias e Consultas via SPARQL**:
   Pretendemos encontrar uma forma de representação do conhecimento a respeito da rede de colaboração que seja útil e possa ser extraído e reutilizado.
3. **Propor uma forma de extração de conhecimento prévio da ontologia**:
   Almejamos propor um método de extração das informações obtidas via consultas SPARQL na ontologia, construção de uma base de conhecimento prévio, e utilização desse conhecimento no enriquecimento do modelo, via adição de novos atributos às entidades da rede de colaboração científica.
4. **Executar a ideia proposta**:
   Desenvolver um algoritmo que recebe como entrada um conjunto de dados sobre pesquisadores, suas publicações e seus coautores, que seja capaz de popular a ontologia e extrair dela o conhecimento prévio, enriquecer o modelo de predição, e executá-lo.
5. **Realizar teste da ideia com informações da Plataforma Lattes**:
   Preparar um ambiente de testes, que contenha um conjunto de dados extraídos da Plataforma Lattes através do *scriptLattes* e a ontologia já estruturada. Depois disso, devemos popular a ontologia e extrair daí conhecimento prévio, rodar o modelo de predição com e sem o enriquecimento e comparar os resultados.

1.4 Contribuições

A principal contribuição que esperamos alcançar com este trabalho é a construção de um modelo que utilize aprendizado supervisionado e seja capaz de receber informação (*background-knowledge*) extraída de uma ontologia e de tratar o problema de predição de ligações em uma rede de colaboração científica de forma eficiente.

O interesse desta contribuição é a proposta de uma forma de representação mais compacta e expressiva do conhecimento prévio, algo que pode permitir uma fácil validação, expansão, generalização e reutilização desse conhecimento, além de uma consequente melhoria da capacidade de predição desse modelo.

1.5 Organização do Trabalho

No Capítulo 2, apresentamos alguns conceitos teóricos que servem para dar um melhor entendimento do trabalho. Já no 3, exploramos alguns trabalhos relacionados, algumas questões que ainda não foram exploradas por outros pesquisadores e justificamos a importância deste projeto de pesquisa. No 4, discutimos um experimento que servirá para implementar e validar as ideias presentes no projeto.

Finalmente, no Capítulo 5, delimitamos o escopo do projeto e discutimos algumas conclusões. O 6 trata do cronograma de atividades necessárias para a execução do projeto.

Capítulo 2

Conceitos

Alguns conceitos teóricos são importantes para o entendimento do presente projeto. Aqui, discutimos os fundamentos mais importantes.

2.1 Aprendizado Supervisionado

O aprendizado supervisionado é um tipo de aprendizado de máquina em que o agente, durante a fase de treinamento, observa alguns exemplos de entradas (*inputs*) e de saídas (*outputs*) esperadas e aprende a classificá-las, através de uma função de transformação de entradas em saídas. Depois disso, esse agente observa uma base de testes com inúmeros novos exemplos e deve classificá-los de acordo com o seu modelo interno. Dizemos que um modelo é eficaz se maximiza o conjunto de observações classificadas corretamente e minimiza as observações classificadas incorretamente.

2.2 Conhecimento prévio (*background knowledge*)

Conhecimento prévio, segundo Russell e Norvig (2009, capítulo 19), geralmente é representado como um conjunto geral de teorias em uma lógica de primeira ordem. Essas teorias são compostas por hipóteses que devem explicar ou classificar corretamente um conjunto de observações (ou atributos). Essas observações são sentenças lógicas que descrevem algo sobre o mundo. As hipóteses devem poder ser generalizadas e devem ser aplicáveis a novos exemplos. Outra propriedade importante é que a teoria seja consistente, ou seja, uma hipótese não pode gerar falsos positivos ou falsos negativos.

Esse conhecimento prévio pode ser cumulativo: novas observações podem gerar novas hipóteses, que enriquecem o conhecimento prévio e o modelo como um todo, tornando mais eficaz a sua capacidade de predição e aumentando a generalidade das hipóteses.

Uma das áreas que se concentra nesse tipo de problema de aprendizado é chamada de *Inductive Logic Programming* (ILP), que é uma intersecção entre Aprendizado de Máquina e Programação Lógica. Os programas lógicos são compostos por fatos ou predicados que descrevem exemplos (observações do mundo), conhecimento prévio, e hipóteses. Partindo disto, o programa deriva de forma indutiva um conjunto de novas hipóteses que tenham como consequência lógica todos os exemplos positivos e nenhum dos negativos. Esse conjunto de hipóteses é também chamado de teoria. As novas hipóteses geradas devem ser consistentes com as teorias já presentes no programa. O interessante dessa técnica é o uso de conhecimento prévio definido de uma forma declarativa e compacta, o reaproveitamento de hipóteses e a garantia da consistência.

Com o uso de conhecimento prévio, é reduzida a complexidade do aprendizado, pois as novas hipóteses geradas devem ser consistentes com as hipóteses anteriores, e isso reduz o conjunto de possibilidades que o algoritmo precisaria considerar, além de conseguir explicar parte das novas observações, através do reaproveitamento das hipóteses encontradas ou declaradas anteriormente (Russell e Norvig, 2009, capítulo 19). Por isso, podemos dizer que problemas expressos de forma

relacional podem ser tratados muito bem com algoritmos de ILP e com o uso de conhecimento prévio.

2.3 Hierarquia de Representações em Problemas de Aprendizado

Segundo Raedt (2008), existem vários tipos de representações que podem ser utilizadas para modelar um problema de aprendizado. Ele mostra em seu trabalho que também existe uma hierarquia dessas representações em uma ordem crescente de expressividade, exibida a seguir:

- **Representações Booleanas** (*Boolean Learning*): cada percepção contém itens ou proposições verdadeiras (ou *presentes*) e falsas (ou *não-presentes*), e esperamos encontrar uma regra que defina essas observações. Esta é a representação com o menor grau de expressividade.
- **Aprendizado por tabelas de valores e atributos** (*Attribute-Value Learning*): o conjunto de percepções ou experiências é apresentado como uma tabela única onde cada linha é um exemplo e cada coluna é um atributo, e desejamos que o modelo aprenda a classificar uma experiência nova ou a prever o valor de um atributo de acordo com os exemplos vistos anteriormente durante o treinamento. Esse tipo de representação é a mais comumente aplicada na literatura.
- **Representações Multi-Instância** (*Multi-Instance Representations*): muito parecida com a representação por valores e atributos, só que, neste caso, classes podem conter múltiplos exemplos, ou seja, um exemplo pode ter um atributo que depende do valor contido em outro exemplo.
- **Aprendizado Relacional** (*Relational Learning*): múltiplos exemplos ou relações entre exemplos e hipóteses podem aparecer e é possível construir essa representação com múltiplas tabelas em um banco de dados. É bastante útil quando o domínio possui uma natureza relacional.
- **Programas Lógicos** (*Logical Programs*): é um modelo que recebe como entrada as diversas observações (exemplos) e aprende a sintetizar um programa lógico com regras gerais (hipóteses) capazes de derivar estes exemplos e classificar outros conjuntos de observações. Existem alguns trabalhos na literatura cuja linguagem de representação utilizada foi o PROLOG e que usavam ferramentas de ILP. Tal representação é a mais expressiva.

Raedt nos mostra ainda que as representações são equivalentes em sua capacidade de representação: um domínio que é representado por uma pode ser adaptado e transformado em outra representação de mais baixo nível e vice-versa. Entretanto, elas não serão equivalentes em sua capacidade de expressão (expressividade): um modelo de mais alto nível será declarativo e compacto, enquanto sua versão de baixo nível vai ser prolixa e vai necessitar de um número muito maior de exemplos, atributos, tabelas, colunas e outras entidades, o que dificulta a análise e a apreciação dessas informações. Portanto, a escolha correta da representação é fundamental para o pesquisador que deseja explorar um problema, pois isto simplifica a análise do domínio.

Raedt também conclui que todas essas representações podem ser modeladas naturalmente com lógica de predicados, que é relacional por natureza. Também dá exemplos de vários algoritmos de aprendizado utilizando programação com lógica, como *SVMs*, *k-nearest neighbors* e *redes bayesianas*, que podem ser utilizados na modelagem do problema.

2.4 Ontologia

Diferentemente da noção filosófica de Ontologia como o estudo da natureza do Ser, para a Ciência da Computação uma ontologia é uma estrutura relacional, um modelo de especificação formal explícita de conceitos de um determinado domínio. Esses conceitos podem dizer respeito a entidades, ou a relações entre entidades desse mesmo domínio.

Segundo Guarino *et al.* (2009), uma ontologia é um artefato computacional que possui um modelo formal de representação da estrutura de um domínio, contendo as entidades relevantes, bem como as relações que emergem da observação desse mesmo domínio, porém utilizando apenas as que são úteis para algum determinado fim. A modelagem de classes e relações entre entidades é feita através de definições formais em uma lógica de descrição, de forma compacta e de alto nível de abstração.

Uma ontologia tem por finalidade responder a algumas perguntas, também chamadas de Questões de Competência. No fundo, são informações sobre algum domínio que desejamos que ela nos responda.

Um exemplo de domínio possível é a comunidade científica, com seus diversos pesquisadores e suas relações com os demais. Uma ontologia que representa esse domínio deve ser capaz de capturar as entidades relevantes, organizando as informações em conceitos e relações, para que seja capaz de responder a algumas perguntas, como por exemplo: *Quem é coautor de um artigo A?* Nesse caso, ela pode descrever classes de entidades como *Pesquisador* e *Artigo* e relações binárias entre dois indivíduos, como a relação *colaborou_com* e *publicou_artigo*.

O mais importante é que essa definição formal possa ser entendida por um computador em um formato padronizado, para que a resposta seja encontrada de forma automática. Para isso, uma ontologia pode utilizar a inferência lógica para responder a esse tipo de pergunta, como também descobrir novas relações entre entidades, através do uso de *reasoners*, que são programas capazes de inferir consequências lógicas a partir de uma base de fatos.

Por tudo isso, podemos dizer que uma ontologia é uma ferramenta bastante poderosa, capaz de descrever um domínio de forma compacta, permitir consultas complexas e derivar novos conhecimentos de forma automática.

Quanto às questões práticas, existem algumas ferramentas interessantes que podem ser utilizadas para se trabalhar com ontologias. Uma delas é o Protégé [1], que é um editor de ontologias *open-source* escrito em Java. O projeto possui uma comunidade bastante ativa, contando com centenas de milhares de usuários, e é utilizado na criação de sistemas inteligentes e na engenharia de ontologias, sendo capaz de armazenar dados em diferentes formatos, como XML, RDF, e até em um banco de dados. Utiliza a OWL-API, uma biblioteca para Java muito utilizada na construção e manipulação de ontologias OWL, descrita por Horridge e Bechhofer (2011).

Outra ferramenta interessante é a Tawny-OWL [2], desenvolvida por Lord (2013), que nada mais é que uma linguagem de domínio específico feita para a construção de ontologias OWL que também tem como suporte a OWL-API. Essa biblioteca foi desenvolvida na linguagem de programação Clojure, que é um dialeto de LISP que roda na Java Virtual Machine (Hickey, 2008). A biblioteca Tawny-OWL possui um conjunto rico de ferramentas extensíveis com um grande poder de abstração. Outra vantagem é a possibilidade do uso de programação paralela e distribuída para o tratamento de grandes volumes de dados, algo que o Clojure suporta muito bem.

2.5 Ontology-Based Data Access

Acesso a Dados baseado em Ontologias (Ontology-Based Data Access) é uma forma de acesso a dados estruturados através do uso de ontologias. A arquitetura desse tipo de solução inclui uma ontologia que descreve um domínio específico, uma ou mais bases de dados, e um mapeamento que liga as classes da ontologia com alguns conjuntos de dados presentes nessas bases de dados. Esse mapeamento transforma consultas SPARQL em consultas na linguagem utilizada pelo SGBD, seja SQL, noSQL, ou qualquer outra. Essa abordagem nasceu para facilitar o acesso e a consulta às bases de dados.

[1] http://protege.stanford.edu

[2] Repositório de código da biblioteca Tawny-OWL: https://github.com/phillord/tawny-owl

Capítulo 3

Trabalhos Correlatos

Neste capítulo, apresentamos algumas contribuições importantes que tratam do problema de análise de redes de colaboração científica e predição de ligações, e outros trabalhos relacionados que serviram de inspiração para este projeto de pesquisa.

3.1 scriptLattes

A Plataforma Lattes é uma plataforma vinculada ao CNPq, e a mais importante base integrada de currículos, grupos de pesquisa e instituições de ensino do Brasil, registrando informações valiosas sobre as atividades de pesquisa e o perfil de pesquisadores de diversas áreas do Saber em todo o país.

O projeto *scriptLattes*, proposto por Mena-Chalco e Junior (2009), é um sistema capaz de fazer mineração dos dados de currículos presentes na Plataforma Lattes e de gerar vários relatórios acadêmicos, além de disponibilizar informações sobre as publicações dos pesquisadores brasileiros, fazendo desambiguação dos autores e artigos e exportando dados sobre coautoria e outros tipos de colaboração.

O nosso trabalho vai utilizar uma base de mais de 4 milhões de currículos extraídos da Plataforma Lattes, gentilmente cedida pelo professor Jesús P. Mena-Chalco, como base de testes para o modelo aqui proposto.

3.2 Predição de Ligações

Vários trabalhos presentes na literatura exploram o problema da predição de ligações (*link prediction*) em redes sociais. Esse problema possui diversas aplicações, como na análise e reconstrução de redes e em sistemas que utilizam informações pessoais para sugerir novos contatos ou novos amigos. Outros pretendem detectar membros de redes terroristas com o intuito de prevenir ataques.

Uma outra aplicação interessante, discutida no item 3.2.4, conseguia encontrar artigos ou documentos relacionados e sugerir citações.

Discutimos a seguir alguns trabalhos relevantes.

3.2.1 Importância de um pesquisador, financiamento, e quantidade de colaboradores em um projeto

Em Ebadi e Schiffauerova (2015), encontramos uma análise muito apurada sobre uma rede de colaboração científica que serviu para levantar os fatores que determinam a importância de um pesquisador nessa rede, sua centralidade e sua produtividade. Foram feitas várias constatações interessantes. Descobriram, por exemplo, que grupos ligados a organizações que produziam muitos artigos importantes acabavam tendo uma performance melhor do que outros grupos.

Outro fato interessante é que a experiência e os anos de trabalho de um indivíduo fazem com que ele seja mais conhecido em sua comunidade. Tendo acesso a dados sobre financiamento e fomento à

pesquisa, constataram que pesquisas fomentadas por indústrias e empresas acabam atraindo mais colaboradores.

Essa análise também foi baseada em medidas feitas na estrutura da rede, como a centralidade dos vértices, *eigenvectors* e coeficientes de agrupamento, numero médio de coautores (ligações) de um dado pesquisador, dentre outros indicadores. Com essas medidas, foi possível encontrar os líderes de diversas comunidades, que são pessoas com grande influencia local.

Descobriram, ainda, que os pesquisadores mais produtivos e com o trabalho de melhor qualidade também eram os mais colaborativos. Também mostraram a influência que o financiamento causa nas redes locais de colaboração ao longo do tempo, fazendo com que os cientistas que recebem financiamento e ocupam posições de liderança busquem coautores em outras comunidades mais distantes, ampliando o seu canal de conexões.

3.2.2 Predição utilizando Aprendizado Supervisionado

Hasan *et al.* (2006) aplicou e comparou a eficácia de diversas técnicas de aprendizado supervisionado, tais como *SVMs*, *Árvores de Decisão*, *Multilayer Perceptrons*, modelos de classificação utilizando kNN (*k-nearest neighbors*), *Naive Bayes*, e *RBF Networks*. Sua aplicação foi na detecção de membros de redes terroristas.

O uso de *SVMs* mostrou-se melhor do que as outras técnicas, por obter uma taxa de acerto um pouco maior e por possibilitar a escolha das características importantes a serem utilizadas pelo modelo através de um método automatizado de filtragem.

O aspecto mais importante desse projeto é que a escolha do algoritmo não parece fazer tanta diferença assim na eficácia do modelo. Podemos, com isso, concluir que uma boa escolha das características é um dos fatores mais importantes.

3.2.3 Grafos Relacionais com Atributos

Um experimento muito interessante e inspirador foi explorado por Cervantes (2014), que modelou a rede de colaboração científica através de grafos relacionais com atributos para mostrar as relações de coautoria entre pesquisadores. Foi também criado um modelo baseado em *SVMs* capaz de prever novas colaborações (ligações) entre pesquisadores a partir de dados de treinamento extraídas da Plataforma Lattes via *scriptLattes*.

O modelo proposto permitiu uma série de outras análises relevantes da estrutura dessa rede, como a identificação de pesquisadores mais importantes, ou mais colaborativos, que correspondem aos vértices com mais conexões. Também foi possível identificar comunidades dentro de diferentes áreas formadas por componentes conexos desse grafo.

Por sua flexibilidade e alto nível de expressividade, esse modelo em grafo com atributos pode ser facilmente expandido. Entretanto, o modelo de aprendizado utilizado recebe como entrada tabelas de atributos e valores. Por causa disso, é preciso transformar uma representação relacional mais expressiva (um grafo) em uma representação mais simples (uma tabela) para que o algoritmo funcione.

3.2.4 Predição de Citações

Uma aplicação de Aprendizado Estatístico Relacional (SRL) foi feita por Popescul e Ungar (2003) em um sistema de predição de citações, que nada mais é do que um modelo de predição de ligações entre documentos. Analisando informações a respeito de duas publicações, o sistema calculava a probabilidade de uma publicação citar a outra. Com isso, o sistema poderia sugerir trabalhos correlatos que pudessem ser citados por um autor durante a escrita de algum artigo ou outro documento.

O grande desafio desse modelo de aprendizado é a seleção de atributos (*features*) que auxiliem na tarefa de classificação. Como o problema está modelado de forma relacional, a quantidade de

atributos pode crescer indefinidamente. É preciso, portanto, encontrar uma forma inteligente de adicionar um atributo e avaliar sua relevância, através de algum tipo de verificação.

Em um trabalho posterior, Popescul e Ungar (2007) sugerem um modelo interessante de seleção de atributos: a geração de novos atributos se dá via busca no conjunto de atributos possíveis, e sua seleção é feita através de um teste que avalia a significância estatística da inclusão desse atributo no modelo de predição.

Aplicado ao mesmo problema de predição de citações, o conjunto de atributos possíveis é derivado de consultas SQL relacionadas às entidades pertencentes a um banco de dados relacional, e o algoritmo explora esse conjunto e seleciona os atributos mais relevantes segundo sua significância estatística. O interessante desse trabalho é que o algoritmo, ao gerar novos atributos, adiciona essas novas informações ao mesmo banco de dados como novas relações entre entidades, enriquecendo ainda mais o modelo.

Nossa proposta é parecida, pois também desejamos gerar novos atributos para essas entidades, explorando algumas consultas possíveis, extraindo conhecimento e formalizando esse resultado como *background-knowledge*, que será usado para enriquecer o modelo de aprendizado. A principal diferença está no uso de uma ontologia em vez de um banco de dados relacional.

3.3 Ontologias

Uma rede social é uma estrutura relacional que podemos facilmente representar por um grafo ou por uma ontologia. Alguns trabalhos modelaram ontologias para tratar de problemas relacionados à Plataforma Lattes e outras redes de colaboração, que discutimos a seguir.

3.3.1 Plataforma Lattes

No caso da Plataforma Lattes, Costa e Yamate (2009) construíram uma ontologia capaz de responder a diversas questões de competência. O usuário consulta a base usando linguagem natural, e a ontologia consegue encontrar a informação desejada, através de inferência. A vantagem dessa metodologia é a facilidade na modelagem, nas ferramentas de inferência, e na expressividade da representação. Galego (2013) também desenvolveu uma ontologia para esse domínio como uma extensão do trabalho anterior, e seu interesse foi gerar relatórios e detectar inconsistências.

3.4 Ontology-Based Data Access

3.5 SQL

Uma das primeiras formas de se trabalhar com ODBA com SGBDs relacionais é feita através da tradução de consultas SPARQL originadas de uma ontologia para SQL, que é a linguagem de destino. Com isto, é possível extrair informações do banco de dados e exportá-las para RDF.

O trabalho de Prud'hommeaux e Bertails (2008) propõe uma forma de expressar dados relacionais como um grafo RDF, e uma álgebra para mapear consultas SPARQL do tipo SELECT sobre o grafo RDF em consultas SQL sobre um conjunto de dados relacionais. O grafo RDF é construído com base na estrutura relacional juntamente com identificadores URI. A álgebra especifica uma função que recebe como entrada um identificador, um esquema relacional e uma query SPARQL nesse mesmo grafo, e gera uma consulta relacional que pode ser executada em um banco de dados, produzindo as mesmas soluções que uma consulta SPARQL executada sobre o grafo stem.

Com essa técnica, um banco de dados relacional pode ser exposto na web semântica e receber consultas SPARQL com a mesma performance de consultas SQL comuns.

por que integrar sparql e sql: - integrar bases de dados diversas - dados que desejamos que sejam interpretados por máquinas estão em bancos de dados relacionais - expor dados na web semântica

problemas: - falta de padronização de técnicas de mapeamento

- armazenamento relacional é mais compacto e eficiente do que as triple stores

BGP: SPARQL basic graph pattern

Um BGP pode ser quebrado em um conjunto de conjunções de triplas, onde cada conjunção é equivalente/se encaixa com/diz respeito aos atributos de uma dada relação no banco de dados. Essas conjunções podem ser expressas como uma única relvar.

-> a SPARQL Basic Graph Pattern (BGP) can be broken down into a set of conjunctions of triple patterns, where each conjunct matches the attributes of a given relation in the database.

Uma consulta SPARQL qualquer, com conjunções, disjunções e opcionais podem ser expressos em uma consulta SQL que pode ser executada com a mesma eficiência que SQL convencional.

Partindo do princípio de que um banco de dados é um conjunto de relações em um esquema (schema). Um esquema define um título e um conjunto de atributos, cada um de um certo tipo de dados. Se o atributo for uma chave estrangeira, então o valor do atributo é algum outro atributo em alguma relação. Se não for uma chave estrangeira, então o atributo é primitivo, isto é, um valor como um número, um texto, ou uma data.

Uma consulta SQL de exemplo pode ser definida assim: *SELECT attrList FROM fromList WHERE whereCondition*

fromList é uma lista de tuplas que mapeiam relações para variáveis, chamadas relvars, que identificam um conjunto múltiplo de tuplas, as tuplas da relação. Uma relvar é formada por uma combinação algébrica das relvars de fromList.

Esse mapeamento necessita apenas dos operadores INNER JOIN e LEFT OUTER JOIN, que são indicados por palavras-chave separando tuplas na fromList. A condição where é uma restrição expressa em termos de atributos dessas relvars (atributos de relvars) na fromList. A relvar agregada é restringida pelas condições expressas no where.

A lista attList seleciona um conjunto de atributos da agregação relvar restrita e os renomeia, criando novos nomes de atributos, e produzindo uma nova relação no esquema, definido por attrList.

O grafo de termos/raízes (*stem graph*) ...

3.6 NoSQL

3.6.1 MongoDB

Um recente trabalho de [1] utiliza NoSQL em aplicações OBDA, propondo uma arquitetura genérica de OBDA aplicado a qualquer tipo de SGBD. A primeira aplicação desta arquitetura foi feita com o MongoDB. Com isso, foi possível mapear uma ontologia com um conjunto salvo em um banco de dados e fazer a conversão da consulta SPARQL em uma consulta MongoDB, eliminando assim a necessidade de se extrair previamente esses dados e transformá-los em RDF ou popular a ontologia antes da consulta.

[15] construiu uma ferramenta de mapeamento de consultas SPARQL em consultas a uma base MongoDB para tornar públicos os dados de uma base legada.

A técnica utilizada consiste em gerar uma base RDF virtual, ou seja, os documentos salvos na base foram expostos em formato RDF. Foi proposto um método de tradução em 2 passos: primeiro, a consulta SPARQL é transformada em uma consulta abstrata usando mapeamentos MongoDB para RDF escritos em uma linguagem intermediária chamada xR2RML. Depois, essa consulta intermediária é transformada em uma consulta MongoDB concreta. Como resultado, concluíram que é sempre possível reescrever uma consulta que produza resultados corretos.

3.7 Justificativa

A principal hipótese a ser testada é se o enriquecimento de um modelo de aprendizado supervisionado com conhecimento prévio extraído de uma ontologia pode aumentar a eficácia desse modelo quando aplicado ao problema de predição de ligações, aqui definido como a colaboração entre pesquisadores em uma rede de colaboração. Essa proposta será comparada com os trabalhos de Hasan *et al.* (2006) e Cervantes (2014).

O uso de uma ontologia é justificado por sua expressividade, algo que simplifica a criação de consultas geradoras de novos atributos para as entidades da rede, e a possibilidade da descoberta de novas características, através do uso de inferência lógica. Também é simples encontrar e representar novos tipos de relações entre entidades, bem como organizá-las em diferentes classes. Outra vantagem secundária é a uniformização do vocabulário a respeito do domínio, e a fácil reutilização desses conceitos e conhecimentos em outras aplicações de domínios semelhantes.

Além disso, os resultados do modelo de predição podem ser aproveitados posteriormente para expandir a própria ontologia inicial. As predições resultantes do modelo podem ser adicionadas à ontologia como novas relações. E isto pode gerar novos atributos úteis e novos tipos de classificação que podem ser extraídos novamente, derivando um conhecimento prévio ainda mais rico.

Escolhemos aplicar esse modelo à Plataforma Lattes por ser um domínio bem utilizado e por termos disponível uma base de currículos bastante extensa, da ordem de 4 milhões de arquivos.

Quanto ao modelo de aprendizado supervisionado, serão utilizados os mesmos que estão presentes na literatura para que se faça uma justa comparação.

Capítulo 4

Desenvolvimento

Neste capítulo, apresentamos o desenvolvimento do projeto e as principais práticas que o norteiam, através da discussão de um experimento exploratório em pequena escala (com uma quantidade pequena de dados) que servirá como estrutura inicial do projeto.

4.1 Estudo do domínio e construção de uma ontologia a respeito da Rede de Colaboração Científica

A proposta consiste em rodar um experimento em pequena escala, de onde extraímos informações de um número pequeno de currículos Lattes, com interesse nas informações básicas dos pesquisadores do Departamento de Computação do Instituto de Matemática e Estatística da USP. Para isso, faremos uso dos seguintes dados: nome, área de atuação, publicações e coautores e local de trabalho ou residência.

Com base nessas informações, modelamos uma ontologia utilizando OWL através do software Protégé. A estrutura básica da ontologia serve para organizar o conhecimento a respeito desses pesquisadores e sua origem acadêmica, suas publicações e os relacionamentos que eles têm com outros. Com isso, podemos consultar essas informações estruturadas acerca da produção bibliográfica e científica dos docentes e alunos através de queries SPARQL. O conjunto de dados gerados por essas queries servirão de base para este experimento.

A ontologia Friend of a friend (FOAF), que descreve pessoas, relações e suas atividades, foi escolhida como ponto de partida devido o fato de ela ter alguns conceitos, classes e termos que nos auxiliam na modelagem da rede de colaboração científica.

As classes mais importantes da ontologia criada para este experimento são:

Grupo define grupos de pessoas (grupos de pesquisa, por exemplo).

Organização define uma organização, que pode ser uma Universidade, por exemplo.

Universidade *(subclasse de **Organização**)* classe das universidades e instituições de ensino e pesquisa.

Instituto *(subclasse de **Organização**)* classe dos institutos de ensino e pesquisa e faculdades.

Departamento *(subclasse de **Organização**)* classe dos departamentos ligados a institutos de pesquisa ou faculdades.

Pessoa classe-pai dos tipos de pessoa.

Pesquisador *(subclasse de **Pessoa**)* define pesquisadores, que são pessoas que publicam artigos e desempenham outras atividades científicas. Podem também ser alunos de cursos de pós-graduação, professores, e outros tipos de pessoas.

Publicação define artigos científicos e outras publicações de revistas, simpósios e outros eventos.

Veículo veículos de publicação de artigos científicos, como revistas, jornais, e eventos como simpósios e conferências.

Local define os locais geográficos.

País *(subclasse de **Local**)* classe dos países.

Cidade *(subclasse de **Local**)* classe das cidades.

Algumas relações entre as entidades acima são importantes. Citamos algumas dessas relações abaixo:

membro_de uma **Pessoa** pode ser membro de um **Grupo**. É o inverso da relação **membro**.

autor uma **Pessoa** pode ser o autor de uma **Publicação**. É o inverso da relação **tem_autores**.

trabalhou_em uma **Pessoa** pode trabalhar ou ter trabalhado em uma **Organização**.

publicou um **Veículo** pode publicar uma **Publicação**, como um artigo. É o inverso da relação **publicado_em**.

publicado_em uma **Publicação** pode ser publicada em um **Veículo** qualquer, como uma revista. É o inverso da relação **publicou**.

tem_autores uma **Publicação** possui um ou mais autores da classe **Pesquisador**. Essa propriedade é importante pois indica uma relação de *coautoria* entre dois pesquisadores.

localizado relaciona uma entidade como **Organização** ou **Veículo** com um local geográfico, como uma **Cidade**.

4.2 Consultas

A ontologia foi populada com as informações de currículos Lattes escolhidos, além de algumas consultas que também fizeram parte da definição da ontologia, sendo estas últimas feitas na linguagem SPARQL. As consultas que nos interessam agora respondem às seguintes questões:

- Qual a área de pesquisa de um pesquisador?
- Quem são os pesquisadores que ele orientou?
- Há quantos anos esse pesquisador trabalha em uma instituição?
- Quem são as pessoas que colaboraram, como coautores, em sua produção científica?

A primeira questão de competência é extraída diretamente do currículo Lattes, quando possível. Trata-se da definição de suas áreas de atuação.

A segunda questão pode ser respondida através da relação *orienta* das instâncias da classe *Professor*. Um professor terá uma lista de orientandos, que, por sua vez, também são pesquisadores. A origem dessas informações é a seção de orientações do currículo Lattes. A terceira pode ser respondida através da análise da atuação profissional em instituições de ensino, pois ela contém o ano de início de atuação.

Por fim, a última questão é obtida fazendo-se uma consulta aos artigos publicados pelo pesquisador, aos trabalhos em eventos e aos demais tipos de produção técnica e outras informações bibliográficas que listam os nomes de outros colaboradores. O grande desafio nesse caso é fazer a desambiguação de nomes.

4.3 Predição de Ligações

Aplicaremos alguns modelos de predição, por exemplo, Árvores de Decisão e Máquinas de Vetores de Suporte (SVM). O primeiro, por ser um modelo simples e ótimo para análise preliminar dos atributos relevantes que devem ser incluídos ou descartados, o que permite uma otimização do conjunto de atributos. Já os SVMs são importantes para nosso experimento, pois os trabalhos relacionados utilizaram esse modelo e obtiveram bons resultados. Outra vantagem é a possibilidade do uso de seleção de atributos (*feature selection*), que também permite uma otimização e simplificação do conjunto de atributos que serão considerados pelo modelo.

Na fase A do experimento será construído um grafo com atributos similar ao descrito por Cervantes (2014), levando em consideração as ligações entre pesquisadores, o grau de cada vértice e outros atributos relacionados à estrutura da rede de colaboração. Esses atributos serão extraídos, gerando um conjunto de vetores de características, que chamaremos aqui de VC. Esse conjunto VC será a entrada do modelo de predição na fase inicial do experimento, que também será enriquecido com conhecimento extraído da ontologia.

4.4 Extração do Conhecimento e Construção do *background-knowledge*

O conhecimento prévio, que chamaremos de BK, será extraído da ontologia através das consultas SPARQL e estruturado em uma matriz da seguinte forma: seja P o conjunto de pesquisadores presentes na ontologia. Cada linha da tabela resultante corresponde a um elemento do conjunto de pares distintos de pesquisadores $Q = \{(a, b) | a \in P \text{ e } b \in P \text{ e } a \neq b\}$, e cada coluna recebe o valor de algum atributo de a, ou de b, ou $R(a, b)$ (uma relação entre a e b). $R(a, b) = 1$ se a relação existir ou for válida, e $R(a, b) = -1$ em caso contrário.

Como exemplo, admita um conjunto $P = \{p_1, p_2, p_3\}$ e algumas relações como $R = \{orienta, coautor\}$. Sabemos que p_1 `orienta` p_2, pois p_1 foi orientador de p_2 em seu doutorado. E também sabemos que p_1 `coautor` p_2 e p_1 `coautor` p_3 e p_2 `coautor` p_3, pois os três publicaram um artigo juntos. Também conhecemos alguns atributos desses pesquisadores: todos eles pertencem à área Ciência da Computação cujo identificador será, digamos, 123. Portanto, a matriz resultante será:

	área	orienta	coautor
$Q(p_1, p_2)$	123	1	1
$Q(p_1, p_3)$	123	-1	1
$Q(p_2, p_3)$	123	-1	1

Tabela 4.1: *Matriz de background-knowledge*

Entretanto, essa matriz pode ter um tamanho muito grande, dependendo da quantidade de pesquisadores. Uma solução possível seria limitar os exemplos, mostrando apenas os pares de pesquisadores que também possuem relação de coautoria. Entretanto, o impacto dessa alteração na eficácia do método ainda precisa ser investigado.

O conjunto de vetores de características VC, gerado na fase A, será copiado na fase B e enriquecido com as informações extraídas do conjunto BK. Chamamos aqui de enriquecimento a adição de novos atributos e relações aos elementos de VC. Cada pesquisador possui um conjunto de atributos relevantes, e esse conjunto será expandido com as informações do conjunto BK, gerando o conhecimento-prévio enriquecido, ou *BK-enriched data* (BK_e).

O modelo de predição receberá como entrada esse conjunto de dados BK_e, separado aleatoriamente em subconjuntos de treinamento e teste. Após o treinamento, durante os testes, o modelo deverá prever se existe ou não uma relação de coautoria a partir dos atributos desse par de pesquisadores através da classificação de cada um dos exemplos. Esse resultado será posteriormente validado com o conjunto original, sendo possível com isso medir a eficiência do método.

4.5 Desenvolvimento e Testes

A estrutura inicial do experimento segue o diagrama da Figura 4.1.

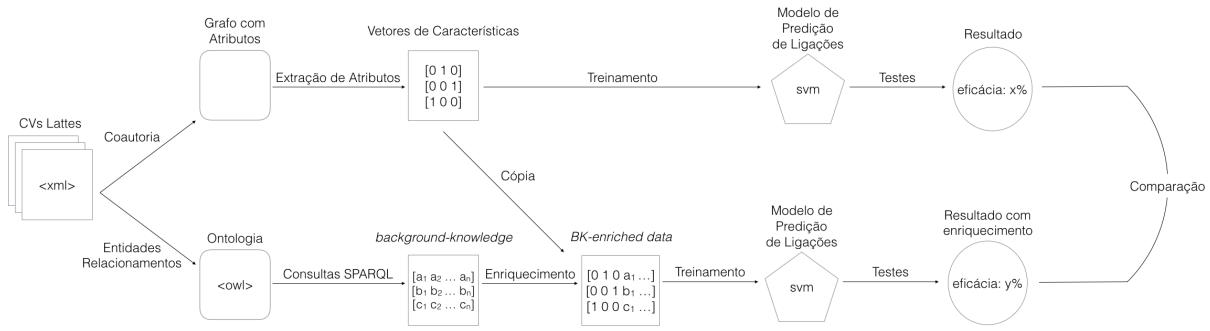

Figura 4.1: *Diagrama de execução do experimento*

Ele está dividido em duas fases, a fase A e fase B. A fase A vai trabalhar apenas com dados relacionados à estrutura da rede, extraídos de um grafo. Já a B vai trabalhar com os dados sobre a estrutura da rede mais os dados extraídos da ontologia, que chamamos de conhecimento prévio. Os passos de cada fase serão discutidos a seguir.

Na fase A, um grupo pequeno de currículos Lattes no formato XML será lido e processado, e serão extraídos dados gerais desses currículos. A partir desses dados, será construído um grafo com atributos, onde cada nó representa um pesquisador, e cada aresta representa alguma relação entre os participantes dessa rede.

Desse grafo, será extraído um conjunto de vetores de características, que chamamos de VC, separado em conjuntos de treinamento e teste. O modelo de predição será treinado com o conjunto de treinamento, e será testado com o conjunto de teste. Essa fase vai gerar um relatório que lista a eficácia do modelo, pois o resultado da classificação pode ser validado com os dados iniciais.

Na fase B, o mesmo grupo de currículos será processado, utilizando esses dados para popular a ontologia descrita anteriormente. Serão feitas consultas a respeito das entidades e relações presentes na ontologia e o seu resultado será transformado em um conjunto de atributos, denominado *background-knowledge*, ou BK, como descrito anteriormente.

Os vetores de características (VC) da fase A serão copiados e associados ao conjunto BK, processo que chamaremos de enriquecimento com conhecimento prévio, gerando o conjunto enriquecido BK_e. Dessa forma, outro modelo de predição será instanciado e receberá como estrada esses dados enriquecidos, também separados em conjunto de treinamento e teste. Será feito o treinamento do modelo, depois o teste deste modelo, seguido de uma análise do resultado.

Por fim, será feita a comparação da eficácia de ambos os modelos, verificando se há alguma melhora no método. Também será avaliado se os atributos adicionais extraídos de BK_e são relevantes para o modelo, através de uma análise da importância relativa de cada atributo na classificação final (*feature selection*).

Em suma: o ambiente de testes recebe um conjunto de currículos e um arquivo OWL correspondente à estrutura da ontologia, porém sem nenhuma instância. O software faz o processamento (fases A e B), e gera ao final um relatório comparando ambas as estratégias.

4.6 onto-mongo

O *onto-mongo*[1]é um sistema desenvolvido para tratar o problema de transformação de consultas SPARQL em consultas ao SGBD noSQL. Ele transforma consultas SPARQL oriundas de alguma Ontologia OWL em consultas ao MongoDB através de um mapeamento. O resultado desta consulta

é transformada em triplas RDF que são usadas pela ontologia.

O protótipo possui as seguintes partes:

- API
- Ontologia
- Mapeamento Ontologia x Coleções do Banco de Dados
- Tradutor de Consultas SPARQL em Consultas NoSQL
- Exportação
- Atualização da Ontologia

4.6.1 API

Uma API, ou Interface de Programação de Aplicações, é uma aplicação web que possui um conjunto definido de requisições e respostas HTTP, e serve para integrações entre sistemas ou serviços diversos. Em nosso caso, a API é um software que recebe mensagens de entrada contendo uma consulta em linguagem SPARQL e responde a essa consulta com um conjunto de triplas RDF extraídas de um banco de dados.

Essa API foi desenvolvida em *ruby*[2]- uma linguagem orientada a objetos - e utiliza o framework *ruby on rails*[3], criado para facilitar a criação de aplicações e serviços web.

O projeto foi desenvolvido utilizando docker, que é uma plataforma aberta com várias ferramentas de virtualização para aplicações distribuídas, dotada dos chamados *containers*, que são pequenas máquinas virtuais linux que são executadas de forma independente. Isto simplifica a configuração dos ambientes de desenvolvimento e de aplicação, que são inicializados segundo um arquivo de configuração, permitindo que sejam definidos os softwares que devem ser instalados em cada máquina virtual e como devem estar configurados os ambientes.

Como exemplo, em nosso projeto foram criados dois *containers*, um para rodar a aplicação web (API) e o servidor, e outro para rodar o banco de dados MongoDB.

A API possui classes escritas na linguagem *ruby*. Essas classes representam coleções presentes no banco de dados. Através destas classes, é possível consultar o banco de dados, através do Mongoid. A API possui duas classes de modelo (*model*) principais: *Researcher*, e *Publication*. A primeira diz respeito às informações do pesquisador, a segunda se relaciona com publicações científicas.

4.6.2 Mapeamento

A API possui classes escritas na linguagem *ruby*, que é uma linguagem orientada a objetos. Essas classes representam coleções presentes no banco de dados onde são guardadas informações a respeito de pesquisadores e suas publicações. Através destas classes, é possível consultar o banco de dados, através do Mongoid.

Nosso mapeamento foi construído para ligar as classes da ontologia com as classes do onto-mongo, para que fosse possível consultar informações relacionadas a essas entidades no banco de dados de forma fácil. Utilizando essa abordagem, uma classe da ontologia é mapeada para uma classe ruby que representa ideias similares. As relações entre classes da ontologia também são mapeadas para relações entre classes da linguagem orientada a objetos.

O mapeamento é feito através da adição de alguns métodos de classe que indicam qual relação entre classes e propriedades da ontologia equivalem a qual atributo na classe ruby, ou a qual relação entre classes da ontologia é equivalente a uma relação entre classes do onto-mongo.

[1]O projeto é de código aberto e o repositório está disponível no github, em: http://github.com/thdaraujo/onto-mongo

[3]The Ruby Language https://www.ruby-lang.org/

[3]Ruby on Rails http://rubyonrails.org/

Com isso, tornou-se possível mapear a estrutura presente na ontologia com a estrutura das coleções presentes no banco de dados, passando pelas classes do projeto, conforme mostrado a seguir:

Incluindo o módulo *OntoMap* (linha 3 de ambas as classes) em uma classe ruby como um *mixin*, que é um tipo especial de herança múltipla, e que provê novas funcionalidades à classe e o acesso aos métodos de mapeamento, passando com isso a fazer parte da tradução. Já o método *ontoclass* (linha 9 de *Researcher*) define qual classe da ontologia está relacionada à classe *Researcher*. No caso, ela se relaciona com a classe *foaf:Person* da ontologia.

O método *maps* recebe dois parâmetros: *from*, que diz respeito a algum relacionamento ou propriedade pertencente a uma classe da ontologia e *to*, que indica qual atributo da classe *Researcher* ou relacionamento com outra classe são equivalentes. Na linha 9, definimos um mapeamento entre a propriedade *foaf:name* e o atributo *name*. Na linha 10, a relação *published* é mapeada para *publications*, que representa o conjunto de publicações de um dado pesquisador. Importante ressaltar que uma instância da classe pesquisador pode possuir uma ou mais publicações.

O mapeamento aqui apresentado permitiu a tradução das consultas SPARQL em consultas MongoDB sem a necessidade de haver um mapeamento direto entre a estrutura da ontologia com as coleções de documentos do banco de dados. Em outros trabalhos da literatura, quando falamos de OBDA para SGBDs relacionais, esse tipo de mapeamento se faz diretamente entre consultas SQL e classes da ontologia. Nossa abordagem possui assim uma camada conceitual que faz a ligação entre ontologia e os dados descrita em uma linguagem orientada a objetos.

4.6.3 Tradutor

O tradutor transforma consultas SPARQL em consultas ao SGBD, utilizando para isso o mapeamento. A consulta SPARQL é primeiro transformada em uma *SPARQL Syntax Expression (S-Expression)* através da biblioteca *sxp*[4], pois isto facilita a tradução. A consulta é então transformada em uma estrutura de dados parecida com uma lista aninhada que contem as triplas RDF, filtros, agrupamentos, e variáveis que devem ser retornadas para esta consulta.

4.6.4 Exportação

Os dados extraídos do banco de dados são transformados em triplas RDF, que são enviadas como resposta à consulta SPARQL inicial. Com isso, é possível popular uma ontologia ou consultar e analisar esses dados de forma agnóstica, isto é, sem que o cliente precise saber da estrutura do banco de dados e como estão armazenadas as informações.

[4]Disponível em: https://github.com/dryruby/sxp.rb

Capítulo 5

Conclusões

Uma ontologia é uma forma mais expressiva de representação de conhecimento, e com ela, podemos gerar várias informações úteis sobre o domínio estudado. O uso dessa ferramenta no estudo das redes de colaboração trará uma melhor utilização do conhecimento e uma melhor exploração do domínio, auxiliando na descoberta de novos significados.

Esperamos, com o modelo proposto no presente trabalho, obter um aumento da eficácia de modelos de predição de ligações, através do uso desse conhecimento extraído da ontologia. Com isso, desejamos avançar o estado-da-arte no tratamento do problema de predição de ligações, e construir algo que possa ser utilizado por outros que se interessem por esse tipo de problema.

Capítulo 6

Cronograma

O cronograma de atividades está descrito na tabela a seguir.

	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev
1	X								
2	X	X							
3		X	X						
4			X	X					
5					X	X	X		
6					X	X	X	X	
7									X

Tabela 6.1: *Cronograma de atividades*

Item 1. Estudar o domínio e modelar uma Ontologia.

Item 2. Modelar as consultas à ontologia que servirão para enriquecer o modelo de predição.

Item 3. Construir um modelo de predição de ligações baseado nos trabalhos da literatura, fazendo experimentos com diferentes técnicas de aprendizado supervisionado.

Item 4. Propor e implementar um algoritmo de extração de conhecimento da ontologia e enriquecimento do modelo de predição.

Item 5. Montar ambiente de testes e elaborar alguns testes de comparação.

Item 6. Analisar os resultados e escrever a dissertação.

Item 7. Defender a dissertação.

Referências Bibliográficas

Botoeva *et al.*(**2016**) Elena Botoeva, Diego Calvanese, Benjamin Cogrel, Martin Rezk e Guohui Xiao. OBDA Beyond Relational DBs : A Study for MongoDB. *Proc. of the 29th Int. Workshop on Description Logics (DL 2016)*, 1. Citado na pág. 12

Cervantes(**2014**) Evelyn Perez Cervantes. Análise de Redes de Colaboração Científica: Uma Abordagem Baseada em Grafos Relacionais com Atributos. Dissertação de Mestrado. Citado na pág. 2, 3, 10, 12, 17

Costa e Yamate(**2009**) Anauê Costa e Fabio Yamate. Semantic Lattes: uma ferramenta de consulta de informações acadêmicas da base Lattes baseada em ontologias, 2009. Citado na pág. 2, 11

Ebadi e Schiffauerova(**2015**) Ashkan Ebadi e Andrea Schiffauerova. How to become an important player in scientific collaboration networks? *Journal of Informetrics*, 9(4):809–825. ISSN 17511577. doi: 10.1016/j.joi.2015.08.002. URL http://dx.doi.org/10.1016/j.joi.2015.08.002http://linkinghub.elsevier.com/retrieve/pii/S1751157715000565. Citado na pág. 9

Fayyad *et al.*(**1996**) Usama Fayyad, Gregory Piatetsky-Shapiro e Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37. Citado na pág. 1

Galego(**2013**) Eduardo Ferreira Galego. Extração e Consulta de Informações do Currículo Lattes baseada em Ontologias. Dissertação de Mestrado. Citado na pág. 11

Guarino *et al.*(**2009**) Nicola Guarino, Daniel Oberle e Steffen Staab. *Handbook on Ontologies*, chapter What Is an Ontology?, páginas 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-92673-3. doi: 10.1007/978-3-540-92673-3_0. URL http://dx.doi.org/10.1007/978-3-540-92673-3_0. Citado na pág. 7

Halevy *et al.*(**2009**) Alon Halevy, Peter Norvig e Fernando Pereira. The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems*, 24(2):8–12. ISSN 1541-1672. doi: 10.1109/MIS.2009.36. URL http://dx.doi.org/10.1109/MIS.2009.36. Citado na pág. 1

Hasan *et al.*(**2006**) Mohammad Al Hasan, Vineet Chaoji, Saeed Salem e Mohammed Zaki. Link prediction using supervised learning. Em *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*. Citado na pág. 10, 12

Hickey(**2008**) Rich Hickey. The Clojure Programming Language. Em *Proceedings of the 2008 Symposium on Dynamic Languages*, DLS '08, páginas 1:1–1:1, New York, NY, USA. ACM. ISBN 978-1-60558-270-2. doi: 10.1145/1408681.1408682. URL http://doi.acm.org/10.1145/1408681.1408682. Citado na pág. 7

Horridge e Bechhofer(**2011**) Matthew Horridge e Sean Bechhofer. The OWL API: A Java API for OWL Ontologies. *Semant. web*, 2(1):11–21. ISSN 1570-0844. URL http://dl.acm.org/citation.cfm?id=2019470.2019471. Citado na pág. 7

Kay(**2014**) Alan Kay. The Future Doesn't Have to Be Incremental, 2014. URL https://www.youtube.com/watch?v=gTAghAJcO1o. DEMO Enterprise 2014. Citado na pág. 1

Lord(2013) Phillip Lord. The Semantic Web takes Wing: Programming Ontologies with Tawny-OWL. *CoRR*, abs/1303.0213. URL http://arxiv.org/abs/1303.0213. Citado na pág. 7

Mena-Chalco e Junior(2009) Jesús Pascual Mena-Chalco e Roberto Marcondes Cesar Junior. scriptLattes: an open-source knowledge extraction system from the Lattes platform. *Journal of the Brazilian Computer Society*, 15(4):31–39. ISSN 0104-6500. doi: 10.1007/BF03194511. URL http://dx.doi.org/10.1007/BF03194511. Citado na pág. 9

Michel *et al.***(2016)** Franck Michel, Catherine Faron-Zucker e Johan Montagnat. A mapping-based method to query MongoDB documents with SPARQL. Em *International Conference on Database and Expert Systems Applications*, páginas 52–67. Springer. Citado na pág. 12

Popescul e Ungar(2003) Alexandrin Popescul e Lyle H. Ungar. Statistical Relational Learning for link prediction. Em *In Proceedings of the Workshop on Learning Statistical Models from Relational Data at IJCAI-2003*. Citado na pág. 10

Popescul e Ungar(2007) Alexandrin Popescul e Lyle H. Ungar. Feature Generation and Selection in Multi-Relational Statistical Learning. Em Lise Getoor e Ben Taskar, editors, *Introduction to Statistical Relational Learning*, chapter 16, páginas 453–475. The MIT Press. Citado na pág. 11

Prud'hommeaux e Bertails(2008) Eric Prud'hommeaux e Alexandre Bertails. A Mapping of SPARQL onto Conventional SQL. páginas 1–9. URL https://www.w3.org/2008/07/MappingRules/SpringerTemplate/rdb2rdf.pdfhttp://www.w3.org/2008/07/MappingRules/StemMapping. Citado na pág. 11

Raedt(2008) Luc De Raedt. *Logical and Relational Learning*. Cognitive Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-20040-6. doi: 10.1007/978-3-540-68856-3. URL http://www.springer.com/us/book/9783540200406http://link.springer.com/10.1007/978-3-540-68856-3. Citado na pág. 2, 6

Russell e Norvig(2009) Stuart Russell e Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edição. ISBN 0136042597, 9780136042594. Citado na pág. 5