

**MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**GUILHERME DOS SANTOS VILLOTE  
RAÍSSA DUTRA DE OLIVEIRA**

**DESCOBERTA DE REGRAS DE ASSOCIAÇÃO ENTRE DATASETS: UMA  
ABORDAGEM INCLUINDO A GERAÇÃO DE REGRAS RARAS**

**RIO DE JANEIRO  
2020**

GUILHERME DOS SANTOS VILLOTE  
RAÍSSA DUTRA DE OLIVEIRA

DESCOBERTA DE REGRAS DE ASSOCIAÇÃO ENTRE DATASETS: UMA  
ABORDAGEM INCLUINDO A GERAÇÃO DE REGRAS RARAS

Projeto de Final de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(es): Maria Cláudia Reis Cavalcanti, D.Sc.

Rio de Janeiro  
2020

©2020

INSTITUTO MILITAR DE ENGENHARIA  
Praça General Tibúrcio, 80 – Praia Vermelha  
Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

dos Santos Villote, Guilherme; Dutra de Oliveira, Raíssa.

Descoberta de Regras de Associação entre Datasets: Uma abordagem incluindo a Geração de Regras Raras / Guilherme dos Santos Villote e Raíssa Dutra de Oliveira. – Rio de Janeiro, 2020.

74 f.

Orientador(es): Maria Cláudia Reis Cavalcanti.

Projeto de Final de Curso (graduação) – Instituto Militar de Engenharia, Engenharia de Computação, 2020.

1. mineração de regras de associação. 2. regras raras. 3. web de dados. 4. RDF. i. Reis Cavalcanti, Maria Cláudia (orient.) ii. Título

**GUILHERME DOS SANTOS VILLOTE  
RAÍSSA DUTRA DE OLIVEIRA**

**Descoberta de Regras de Associação entre Datasets:  
Uma abordagem incluindo a Geração de Regras Raras**

Projeto de Final de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(es): Maria Cláudia Reis Cavalcanti.

Aprovado em Rio de Janeiro, 26 de outubro de 2020, pela seguinte banca examinadora:

---

Prof. **Maria Cláudia Reis Cavalcanti** - D.Sc. do IME - Presidente

---

Prof. **Ronaldo Ribeiro Goldschmidt** - D.Sc. do IME

---

Prof. **Luiz Carlos Guedes** - D.Sc. do IME

Rio de Janeiro  
2020

*A todos que de alguma forma sempre estiveram presentes na nossa caminhada compartilhando momentos e experiências, e ao Instituto Militar de Engenharia, a quem devemos nossa formação.*

## AGRADECIMENTOS

Agradecemos a todas as pessoas que nos incentivaram, apoiaram e nos motivaram, permitindo com que chegássemos até o final desta longa jornada.

Nossos familiares, principalmente nossos pais, pela paciência e compreensão.

Amigos e mestres.

Em especial a Professora Orientadora Dr. Maria Claudia Cavalcanti, por sua disponibilidade e atenção fora do comum para nos ajudar sempre que houve dificuldades e dúvidas, e ao Dr. Ronaldo Goldschmidt por oferecer a sua experiência e vivência no assunto e nos ajudar a melhorar cada vez mais o nosso conteúdo.

*“First, think. Second, believe. Third, dream.  
And finally, dare.” – Walt Disney*

## RESUMO

Esse trabalho está estreitamente relacionado à necessidade atual de lidar com a grande quantidade de dados presentes na Web de dados, de forma que seja possível geri-los e utilizá-los para extrair informações úteis. Para que seja possível atingir esse objetivo, muito tem sido feito a fim de padronizar a forma de representação desses dados. Iniciativas como a Web Semântica e o formato RDF tornaram os dados inteligíveis tanto por máquinas quanto por humanos, além de facilitar a interoperabilidade entre os diversos conjuntos de dados disponibilizados publicamente na Web.

Várias ferramentas foram desenvolvidas priorizando a mineração de regras de associação entre os dados de um banco. Neste trabalho foram estudados o algoritmo MRAR e a ferramenta MRAR+, onde esta última tem como objetivo buscar enriquecer um *dataset* alvo a partir de *datasets* externos no âmbito da Web de Dados.

O objetivo principal deste trabalho é apresentar uma ferramenta baseada em uma nova técnica de mineração de regras raras combinada com o algoritmo MRAR e com a técnica de enriquecimento do *dataset* apresentada pelo MRAR+. Essa nova técnica de mineração se propõe a identificar regras raras que costumam ser ignoradas pela maioria dos algoritmos disponíveis no mercado devido ao conceito de suporte mínimo que eles utilizam. Assim, uma solução para o problema de regras raras foi identificada e implementada junto com a abordagem do MRAR+ em Java. Além disso, foram realizados dois estudos de caso a fim de validar a solução apresentada e mostrar uma aplicação prática para a mesma. O principal deles utilizou um banco de dados de médio porte sobre Botânica, cujos dados foram extraídos de um banco de dados real, pertencente ao Instituto Jardim Botânico do Rio de Janeiro.

**Palavras-chave:** mineração de regras de associação. regras raras. web de dados. RDF.

## ABSTRACT

This work is closely related to the current need to deal with the large amount of data present on the Web in order to manage that data and use it to extract useful information. In order to achieve this goal, much has been done to standardize the way in which these data are represented. Some initiatives include the creation of the Semantic Web and the RDF format whose main focus is to make data intelligible to both machines and humans and allow interoperability between the various publicly available datasets.

Several tools have been developed prioritizing the mining of association rules between a bank's data. In this work, the algorithm MRAR and the tool MRAR + were studied, where the latter also aims to enrich a target dataset based on data from external datasets within the scope of the Data Web.

The main objective of this work is to present a new rule mining technique associated with the dataset enrichment technique presented by MRAR +, aiming to identify rare rules that are usually ignored by most of the algorithms available on the market due to the concept of minimal support they use. For this, a solution to the problem of rare rules is identified and implemented. Moreover, two case studies were performed, aiming at the validation of the presented solution and at the demonstration of its practical applicability. The main case study used a medium size dataset that contains botanical data extracted from a real database.

**Keywords:** association rule mining. rare rules. web of data. RDF.

# LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de uma visualização em Grafo de um recurso RDF . . . . .	21
Figura 2 – Exemplo de um recurso RDF serializado em N-Triples . . . . .	21
Figura 3 – Nuvem de <i>Datasets</i> do LOD, retirada de (1) . . . . .	23
Figura 4 – Exemplo de um grafo direcionado (com arestas rotuladas). Subgrafo extraído de (2). . . . .	25
Figura 5 – Estrutura de um Item, extraída de (2). . . . .	26
Figura 6 – Estrutura de uma <i>ItemChain</i> . . . . .	26
Figura 7 – Processo de Trabalho do MRAR, extraída de (2) . . . . .	29
Figura 8 – Visão Geral do MRAR+, retirada de (3) . . . . .	30
Figura 9 – Esquerda: Classes de equivalência raras de um <i>dataset D</i> com diferentes suportes mínimos; Direita Superior: Classes de equivalência raras de um <i>dataset D</i> com suporte mínimo igual a 4; Centro da Direita: regras mRG exatas em D com suporte mínimo igual a 4. Recorte de uma Figura extraída de (4) . . . . .	35
Figura 10 – Visão Geral da Abordagem MONET baseada no <i>Business Process Model and Notation</i> (BPMN) . . . . .	38
Figura 11 – Estrutura <i>EntityInfo</i> . . . . .	39
Figura 12 – <i>Dataset Dt_Futebol</i> baseado no <i>Dt_Neymar</i> de 3.1 . . . . .	50
Figura 13 – Modelo em grafo do JabotG. Fonte: (5) . . . . .	54
Figura 14 – Modelo em grafo do JabotG adaptado . . . . .	55
Figura 15 – Localizações de ocorrências da espécie de planta <i>Vanhouttea lanata</i> , extraída de (6). . . . .	57
Figura 16 – Localizações de ocorrências da espécie de planta <i>Orthophytum diamantinense</i> , extraída de (6). . . . .	58
Figura 17 – Localizações de ocorrências das espécies de plantas, extraídas de (6). . . . .	59
Figura 18 – Localizações de ocorrências dos gêneros de plantas, extraídas de (6) . . . . .	60
Figura 19 – Mapa de Climas do Brasil por Alvares et al.(7) retirado de (7) . . . . .	74

## LISTA DE TABELAS

Tabela 1 – Resultado da execução do MONET Tool com o MRAR no conjunto de dados do Dt_Futebol . . . . .	51
Tabela 2 – Resultado da execução do MONET Tool com o MRARE no conjunto de dados do Dt_Futebol . . . . .	52
Tabela 3 – Resultado da execução do MONET Tool com o MRARE no conjunto de dados do JabotG . . . . .	57
Tabela 4 – Resultado da execução do MONET Tool com o MRARE no conjunto de dados do JabotG . . . . .	58

## LISTA DE ABREVIATURAS E SIGLAS

WWW	<i>World Wide Web</i>
URI	<i>Uniform Resource Identifier</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTML	<i>Hypertext Markup Language</i>
DEER	<i>Data Extraction and Enrichment Framework</i>
FOX	<i>Federated Knowledge Extraction Framework</i>
IME	Instituto Militar de Engenharia
FOAF	<i>Friend of a Friend</i>
LIMES	<i>Link Discovery Framework for Metric Spaces</i>
MIS	<i>Minimum Item Supports</i>
MRAR	<i>Mining Multi-Relation Association Rules</i>
MRARE	<i>Mining Multi-Relation Association Rare Rules</i>
MSApriori	<i>Multiple Support Apriori</i>
RSAA	<i>Relative Support Apriori Algorithm</i>
mRG	Gerador Raro Mínimo
mRI	<i>Itemset Raro Minimal</i>
NER	<i>Named-Entity Recognition</i>
RDF	<i>Resource Description Framework</i>
LOD	<i>Linked Open Data</i>
W3C	<i>World Wide Web Consortium</i>
YAGO	<i>Yet Another Great Ontology</i>
BPMN	<i>Business Process Model and Notation</i>
MONET	<i>Mining For Interlinking Web Datasets</i>

## LISTA DE SÍMBOLOS

Conf              Confiança

Supp             Suporte

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	MOTIVAÇÃO	15
1.2	OBJETIVO	16
1.3	JUSTIFICATIVA	16
1.4	METODOLOGIA	17
1.5	CONTRIBUIÇÕES ESPERADAS	17
1.6	ESTRUTURA	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>19</b>
2.1	WEB SEMÂNTICA	19
2.1.1	<i>RESOURCE DESCRIPTION FRAMEWORK (RDF)</i>	20
2.1.2	ONTOLOGIA	20
2.1.3	BANCO DE DADOS E LINGUAGEM DE CONSULTA SPARQL	21
2.1.4	<i>LINKED DATA</i>	22
2.2	DEFINIÇÕES RELACIONADAS À MINERAÇÃO DE REGRAS	22
2.2.1	MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO	22
2.2.2	PROPRIEDADES DOS <i>ITEMSETS</i>	24
2.2.3	REGRAS DE ASSOCIAÇÃO DE MULTIRRELAÇÃO EM GRAFOS DIRECIONADOS	25
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>28</b>
3.1	MRAR+	28
3.2	MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO COM MÚLTIPLOS SUPORTES MÍNIMOS	30
3.3	ENCONTRANDO <i>ITEMSETS</i> RAROS E AS SUAS REGRAS DE ASSOCIAÇÃO RARAS	32
<b>4</b>	<b>MONET: UMA NOVA ABORDAGEM PARA A DESCOBERTA DE ASSOCIAÇÕES ENTRE DATASETS DA WEB DE DADOS</b>	<b>36</b>
4.1	VISÃO GERAL	37
4.2	ALGORITMO MRARE	39
4.3	MONET TOOL: IMPLEMENTAÇÃO DA ABORDAGEM MONET	44
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS</b>	<b>48</b>
5.1	EXPERIMENTO COM UM DATASET SINTÉTICO	49
5.2	EXPERIMENTO COM O JABOTG	53
5.2.1	ESPÉCIES ESPECIALISTAS	56

5.2.2	ESPÉCIES GENERALISTAS E GÊNEROS . . . . .	58
5.2.3	CONCLUSÃO SOBRE O EXPERIMENTO COM O JABOTG . . . . .	60
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>62</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>64</b>
	<b>A – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRAR EM DT_FUTEBOL . . . . .</b>	<b>66</b>
	<b>B – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRARE EM DT_FUTEBOL . . . . .</b>	<b>68</b>
	<b>C – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRAR EM DT_JABOTG . . . . .</b>	<b>70</b>
	<b>D – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRARE EM DT_JABOTG . . . . .</b>	<b>72</b>
	<b>A – MAPA DE CLIMAS DO BRASIL . . . . .</b>	<b>74</b>

# 1 INTRODUÇÃO

## 1.1 Motivação

A quantidade de dados disponibilizados aos usuários da internet tem crescido consideravelmente nos últimos anos. Um problema a ser enfrentado, porém, é a falta de integração entre esses dados. Muitas informações estão relacionadas entre si e essas relações são perdidas por estarem localizadas muitas vezes em diferentes bancos de dados e até em diferentes sistemas.

É necessário, então, investir no desenvolvimento de ferramentas que possibilitem a integração e a interoperação desses dados. Essas ferramentas devem fornecer métodos para gerenciar e utilizar esses dados, estruturando-os e possibilitando a extração de conhecimento e informações úteis.

O número de dados publicados na Web de Dados atingiu mais de 25 bilhões de triplas, sendo que apenas 5% dessas triplas estão ligadas entre diferentes bases de conhecimento segundo (8). Sendo assim, se faz necessário buscar relações perdidas entre essas bases e usá-las, por exemplo, para enriquecer o banco de dados fonte adicionando informações de uma base à outra levando em consideração a natureza da ligação entre os *datasets*.

Além disso, ferramentas da Web de Dados como (9), (10) e (11) foram desenvolvidas na universidade de Paderborn na Alemanha por um grupo especializados na Web Semântica chamado grupo DICE e são utilizadas em conjunto com o objetivo de enriquecer os *datasets* nesse contexto. O IME é um outro exemplo de instituição acadêmica que ganhou destaque nessa área, se preocupando com o rumo da Web de Dados no Brasil ao desenvolver sua ferramenta própria, o MRAR+. Além disso, o corpo docente do IME mantém estreitas relações com o grupo DICE e com iniciativas da Web Semântica como a DBpedia-pt.

Contudo, enriquecer um *dataset*, considerando que cada recurso presente no mesmo possui pelo menos uma ligação externa, pode se tornar muito custoso e -talvez- até mesmo inviável de se realizar. Para isso, foi desenvolvido durante uma tese de mestrado no IME, o MRAR+, uma ferramenta capaz de encontrar regras frequentes no *dataset* em questão e usá-las para selecionar quais dos seus recursos serão enriquecidos.

Durante a utilização do MRAR+, foi identificada uma limitação relacionada a geração de regras de associação de multirrelação devido a utilização de uma abordagem que contempla somente as regras frequentes, com um suporte maior que o mínimo inserido pelo usuário. Essa limitação faz com que regras sempre verdadeiras, que poderiam ser de

grande valia para o conhecimento acerca do conjunto de dados, sejam ou ofuscadas pelas demais, por possuir um suporte menor, ou nem mesmo geradas, já que o seu suporte não atingiu o suporte mínimo. Além disso, ao focar-se em regras frequentes, a análise está sendo feita considerando o conhecimento genérico do *database*, presente no seu *schema*, e não o conhecimento específico que contempla as particularidades/singularidades de seus recursos e das ligações entre eles.

## 1.2 Objetivo

O objetivo desse trabalho é desenvolver uma nova ferramenta baseada no funcionamento do MRAR+, porém, implementando recursos que supram a deficiência existente no algoritmo de mineração MRAR, que foi exposta na subseção 1.1. A ferramenta construída deverá ser aplicada de maneira eficaz na procura por regras de associação de multirrelação que possuem máxima confiança, ainda que apresentem baixo suporte, além daquelas que o suporte e a confiança sejam maiores que um valor mínimo especificado, devendo também ser eficiente no enriquecimento do *dataset* alvo com dados externos da Web Semântica.

## 1.3 Justificativa

A maior parte das pesquisas relacionadas ao assunto de mineração de regras de associação de multirrelação e enriquecimento de *datasets* utiliza os conceitos de suporte mínimo para restringir o espaço de procura das regras e tornar o processo computacionalmente viável. Porém, existem vários bancos de dados com características diferentes das que são consideradas como premissas para o sucesso dessa abordagem.

Bancos com entidades cuja frequência varia de maneira considerável sofrem com um problema chamado de "problema do item raro", e várias regras que poderiam fornecer informações valiosas sobre o banco como um todo são perdidas ao longo da mineração.

O desenvolvimento de um novo algoritmo que não esteja sujeito às limitações do suporte mínimo para gerar regras pode ser de extrema valia em, por exemplo, bancos de dados de lojas para determinar o comportamento de clientes que envolvam a compra de itens caros e duráveis, ou em bancos de dados médicos para determinar, por exemplo, o comportamento de algumas doenças raras. Além disso, esse trabalho pretende consolidar ainda mais o Instituto Militar de Engenharia na vertente de pesquisa relacionada a dados, sua estrutura e extração de informações referentes à Web Semântica.

## 1.4 Metodologia

Os pontos de partida desse projeto foram o artigo "MRAR: *Mining Multi-Relation Association Rules*", de Ramezani, Saraee e Nematbakhsh(2), e a tese "Mineração de Regras de Associação de Multirrelação em Datasets na Web de Dados" feita por Oliveira(3), sendo realizado, primeiramente, um estudo da estrutura do MRAR+.

Após essa etapa, foi feita uma análise bibliográfica destinada à familiarização com a abordagem de outros artigos, teses e livros sobre o assunto de mineração de regras de associação raras e à análise das possíveis soluções presentes neles para a deficiência que foi encontrada durante a utilização do MRAR+, ou para questões parecidas com essa, para então, que fosse possível analisá-las e especificar o algoritmo que melhor se adequa a proposta do MRAR+, e, a partir disso, foi possível implementá-lo.

Por último, foi realizado um estudo de caso cuja primeira parte consistiu em um experimento que utilizou um banco de dados sintético e reduzido criado especificamente para fazer uma comparação entre os resultados obtidos pela utilização do MRAR e do MRARE juntamente com a abordagem da realização de consultas externas presente na ferramenta implementada. Com isso, foi possível testar e validar hipóteses que deveriam ser verificadas em condições controladas com a utilização da nova ferramenta desenvolvida.

Na segunda parte do estudo de caso, foi utilizado um *dataset* baseado no banco de dados real e de grande porte chamado Jabot, com dados legítimos provenientes do Jardim Botânico do Rio de Janeiro e que reflete conhecimentos específicos adquiridos sobre coleções de botânica segundo (12) e (5). Esse experimento serviu para exemplificar e validar o funcionamento da solução proposta frente a uma situação real de utilização.

## 1.5 Contribuições Esperadas

Ao final do trabalho, é esperado que ele contribua de maneira significativa na formalização de conceitos associados a regras raras dentro da visão de grafos já que hoje em dia a maior parte das definições formais são relacionadas a bancos relacionais. Além disso, deseja-se a apresentação de uma ferramenta funcional tanto na tarefa de enriquecimento de *datasets* quanto na tarefa de mineração de regras de associação de multirrelação frequentes e raras, sendo capaz de identificar regras que envolvam aqueles itens que não são frequentes no banco de dados, ou seja, itens raros.

## 1.6 Estrutura

Esse texto tem como objetivo documentar todos os passos desde a concepção do projeto até a sua finalização, e seguirá a estrutura discriminada a seguir:

O capítulo 2 contemplará a fundamentação teórica para este trabalho, contendo informações que são essenciais para o desenvolvimento dele. A seção 2.1 aborda os conceitos de Web Semântica, formato RDF, Ontologias, Banco de Dados, Linguagem de Consulta SPARQL e *Linked Data*. Já a seção 2.2 irá explicitar tanto a mineração de regras de associação de multirrelação quanto os conceitos matemáticos importantes para a concepção da solução que será apresentada como resultado desse projeto, bem como uma formalização dos conceitos usados.

O capítulo 3 irá apresentar trabalhos relacionados ao tema deste projeto e as abordagens proposta por eles para a solução do problema discutido aqui.

O capítulo 4 descreverá a abordagem adotada neste trabalho contemplando a visão geral do funcionamento do protótipo a ser desenvolvido e do algoritmo proposto para a mineração de regras raras.

O capítulo 5 tem como objetivo apresentar um caso de uso com experimentos realizados em *datasets* sintético e real com a ferramenta desenvolvida.

Já o capítulo 6 irá apresentar a conclusão do projeto, fazendo um resumo dos resultados atingidos bem como propostas de melhorias futuras e aplicações possíveis para o que foi desenvolvido.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo iremos apresentar conceitos e termos que são de suma importância para o entendimento do presente trabalho, bem como dos trabalhos relacionados que serviram como motivação e/ou cujas abordagens foram utilizadas para conceber a nossa solução.

### 2.1 Web Semântica

Inicialmente a *World Wide Web* (WWW), ou Web, foi projetada para que homens e máquinas fossem capazes de entender os dados que fossem nela disponibilizados, conforme (13), sendo um sistema de informação onde documentos e outros recursos Web são identificados unicamente por uma *Uniform Resource Locator (URL)*. O problema é que ao contrário da proposta inicial, as informações acabaram sendo disponibilizadas apenas para a leitura humana.

A Web, como conhecemos, é estruturada na forma de documentos, como páginas em *Hypertext Markup Language (HTML)*, onde esses são conectados uns aos outros sem uma semântica clara o suficiente para serem legíveis às máquinas. Uma consequência imediata pode ser observada nas plataformas de busca, como o Google, que faz a busca através das palavras e não da semântica contida nelas, tendo resultados duvidosos muitas vezes.

Tim Berners-Lee propôs, em 2001, uma extensão da Web atual chamada de Web Semântica, também conhecida como Web de Dados, cujo objetivo é tornar as informações contidas na Web comprehensíveis também para o computador, e para que isso fosse possível, o conteúdo da mesma deveria ser reestruturado. Essa extensão especifica uma nova estrutura que indica as relações entre objetos e a informação semântica dessas relações de acordo com (14).

A estrutura padrão escolhida para que os dados sejam disponibilizados é o RDF que é encontrável e tratável a partir de ferramentas desenvolvidas para a Web Semântica. Essa nova estruturação possibilita a interligação entre os dados garantindo a interoperabilidade entre os mesmos, além de tornar os dados legíveis para as máquinas como já foi mencionado anteriormente.

A *World Wide Web Consortium (W3C)* é responsável por estabelecer todos os padrões relacionados aos dados da Web de Dados, desde o formato até a linguagem de consulta no banco da Web Semântica. Além disso, os relacionamentos entre os objetos são definidos por um conjunto de ontologias chamado de vocabulário cuja importância envolve

evitar ambiguidade entre diferentes conjuntos de dados.

### 2.1.1 Resource Description Framework (RDF)

Com o objetivo de implementar a Web Semântica, foi recomendado pela W3C que seja utilizado o RDF para a descrição conceitual ou modelagem de informação de recursos *web*. Esse formato é baseado na ideia de afirmações sobre recursos em expressões do tipo sujeito-predicado-objeto, conhecido por triplas, possuindo diversos formatos de serialização, sendo o usado aqui o N-Triples, variando o formato de codificação de recursos e triplas de formato para formato.

Segundo (15), as triplas são formadas de sujeito, predicado e objeto, sendo o principal objetivo das mesmas especificar alguma propriedade do sujeito. Por sujeito se comprehende o recurso que receberá alguma propriedade. Já o predicado explicita alguma propriedade utilizando uma padronização existente, mais adiante explicada como ontologia. Por fim, o objeto especifica a propriedade que será atribuída ao recurso/sujeito, podendo ser tanto um outro recurso ou, até mesmo, um literal, ou seja, *string*.

Vale ainda ressaltar que, segundo o primeiro princípio exposto pelo Tim Berners-Lee para boas práticas da Web Semântica, é necessário usar *Uniform Resource Identifier* (URI) para nomear coisas/recursos. Pelos segundo e terceiros princípios, deve-se usar *Hypertext Transfer Protocol* (HTTP) para a procura por informações úteis dos recursos em questão através de *browsers*. Por fim, o quarto e último princípio especifica que sejam incluídas em recursos, propriedades que tenham como valor outros URIs, para que as pessoas que forem pesquisar esses recursos, possam descobrir mais informações ao clicar nesses links.

A Figura 1 apresenta uma visualização em forma de grafo enquanto a Figura 2 apresenta as informações de um arquivo no formato RDF, com serialização N-Triples. Percebe-se que o objeto pode ser um recurso URI ou literal e que o formato N-Triples termina sempre a linha com um ponto final. É possível substituir uma parte do URI presente no recurso por um prefixo, como por exemplo, "http://exemplo.com/rafael" por simplismente "ex:rafael", especificando no formato serializado do RDF tal prefixo utilizado, conforme a serialização escolhida.

### 2.1.2 Ontologia

As ontologias, ou vocabulários, definem conceitos e relações que são usadas para descrever uma área de interesse na Web Semântica, segundo o site da (16) (*World Wide Web Consortium*). Ou seja, elas são desenvolvidas para padronizar as relações usadas em situações particulares, como por exemplo na descrição de uma pessoa, fornecendo definições de suas relações, as restrições de suas aplicações e até mesmo de cardinalidade, tendo a sua principal finalidade organizar dados em informações e conhecimentos.

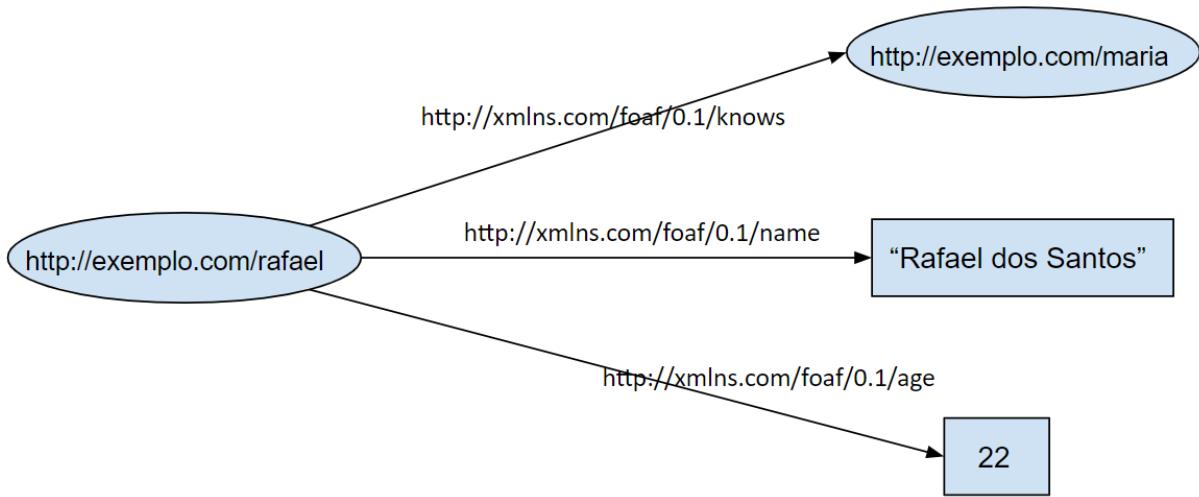


Figura 1 – Exemplo de uma visualização em Grafo de um recurso RDF

`http://exemplo.com/rafael http://xmlns.com/foaf/0.1/knows http://exemplo.com/maria .  
 http://exemplo.com/rafael http://xmlns.com/foaf/0.1/name "Rafael dos Santos".  
 http://exemplo.com/rafael http://xmlns.com/foaf/0.1/nick "Rafa" .`

Figura 2 – Exemplo de um recurso RDF serializado em N-Triples

Essas ontologias da Web Semântica ajudam a integrar os dados de diferentes *datasets*, uma vez que eliminam a ambiguidade que pode existir entre as relações usadas nesses *datasets*, caso ambos usem ontologias padronizadas. Assim, a W3C sugere uma lista de ontologias existentes para serem usadas e, assim, evitar ambiguidades e que várias ontologias distintas sejam utilizadas ao mesmo tempo, ao se criar um padrão na comunidade científica.

Um exemplo de ontologia existente é a (17), sendo uma linguagem de computação definindo um dicionário de termos sobre relações de pessoas que podem ser usados em dados estruturados. Com os termos definidos nela, como mostram a Figura 1 e a Figura 2, é possível definir algumas relações envolvendo pessoas, como, por exemplo, o nome delas, suas idades ou, até mesmo, quem elas conhecem.

### 2.1.3 Banco de Dados e Linguagem de Consulta SPARQL

Ao passo que os dados disponíveis em RDF vêm crescendo, foi preciso um banco de dados capaz de armazenar as suas triplas e gerenciá-las. No âmbito do código aberto, destaca-se o (18), desenvolvida em Java, para aplicações que necessitem de Bancos de Dados na Web Semântica, ele é capaz de armazenar e manipular grafos em RDF nos mais diversos tipos de serialização, possuindo uma interface intuitiva para se trabalhar com

triplas RDF.

Para acessar as informações presentes nesses bancos de dados, é utilizada a linguagem de consulta SPARQL que foi padronizada pela W3C sendo -portanto- confiável. Com essa linguagem, é possível acessar todos os dados de um ou mais grafos em RDF, possuindo funções de união, negação, filtros, agregação, subconsultas, entre outras. Por fim, o resultado dessa consulta pode ser um grafo conexo ou não conexo.

### 2.1.4 *Linked Data*

A Web Semântica definida acima, onde os dados e *datasets* possuem estrutura e ontologias padronizadas, ainda não é o suficiente para fazer a Web de Dados possível. Para isso, é preciso que se tenha uma grande quantidade desses *datasets* disponíveis na Web num formato padrão, alcançáveis e manipuláveis por ferramentas da Web Semântica, segundo o site da W3C(16).

A nuvem *Linked Open Data* (1) é uma iniciativa a fim de integrar os dados de diversos *databases* que são *Open Data*, podendo ser usados livremente e distribuídos por qualquer pessoa, e interligados na Web de Dados. Com isso, é criado uma grande nuvem de dados LOD de diversas fontes, todas em formato RDF, criando a interoperabilidade entre diversas fontes de dados, como é mostrado na Figura 3, onde cada círculo representa um *database* distinto presente na nuvem do LOD.

Um grande exemplo quando se fala de *Linked Data* é a DBpedia(19), que, em linhas gerais, é formada pelos dados da Wikipedia em formato RDF, estando presente no LOD. Contudo, a sua importância não se restringe a isso, uma vez que esse *dataset* possui inúmeras ligações com outros *datasets*, como o GeoNames(20), representando uma referência a ser seguida pelos demais. Além disso, uma variante brasileira, a DBpedia-pt(21), contém os dados da Wikipedia brasileira também no padrão RDF.

## 2.2 Definições Relacionadas à Mineração de Regras

Os artigos (2), (22) e (4) trazem algumas definições de extrema importância para o entendimento do processo de mineração de regras de associação multirrelação. Esses conceitos são descritos a seguir.

### 2.2.1 Mineração de Regras de Associação

A mineração de regras de associação em *datasets* é uma ferramenta muito poderosa e que pode, por exemplo, ser aplicada em bancos de dados referentes a registros de compras de drogarias e mercados com o objetivo de se descobrir as preferências dos clientes. A fim

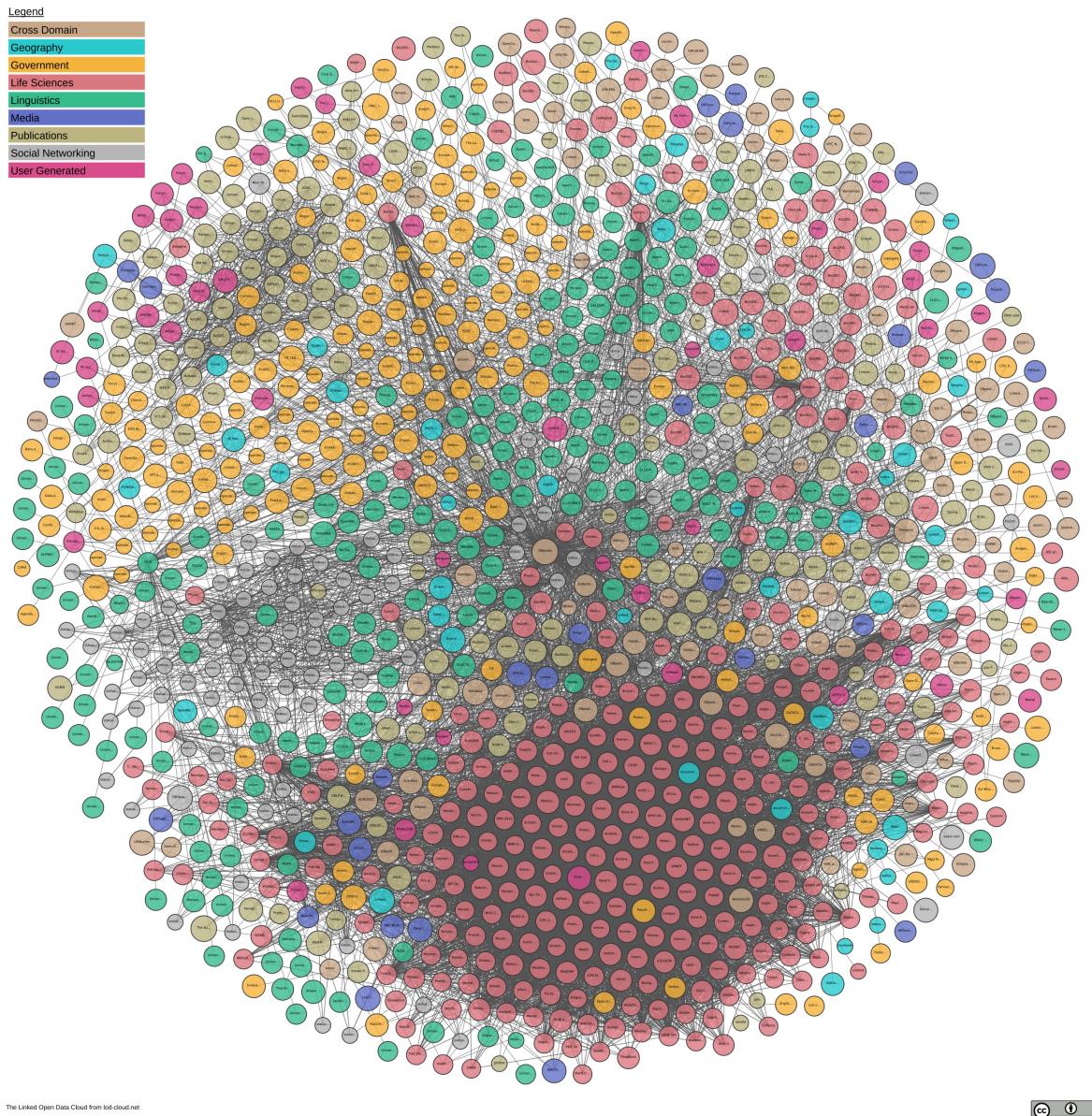


Figura 3 – Nuvem de *Datasets* do LOD, retirada de (1)

de explicar o que é uma regra de associação, é preciso primeiro denotar um item como  $I_i$ , sendo o conjunto de todos os itens do banco  $I = \{I_1, I_2, \dots\}$  referido como um *Itembase*.

Definimos uma transação do Banco de Dados  $T_i$  como um subconjunto de itens e, com isso, um banco de dados é o conjunto de todas as transações  $D = \{T_1, T_2, \dots\}$ . Note que como está sendo considerado que o conjunto de dados possui somente uma relação, como "comprar", por exemplo, então a relação não é apresentada nos itens e nem nas transações acima.

Uma regra de associação  $r$  é definida como  $r : X \rightarrow Y$ , onde  $X$  e  $Y$  representam conjuntos de itens chamados de *ItemSets*, ou *ItemSets* largos quando se tem mais de um item presente, ou seja, subconjuntos de  $I$ , sendo  $X$  chamado de antecedente e  $Y$  de consequente da regra. O suporte de um *ItemSet*  $K$  é o número de transações de  $D$

que contém os itens presentes no *ItemSet* em questão, definido matematicamente como:  $Supp(K) = \frac{|S|}{|D|}$ , onde S é o subconjunto de transações de D que possuem como elementos os itens de K.

Logo, o suporte da regra  $r : X \rightarrow Y$  acima é definido em termos do suporte da união dos *ItemSets* X e Y como sendo  $Supp(r) = Supp(X \cup Y)$ . Por conseguinte, a confiança dessa regra r é definida como a fração dos itens presentes no *ItemSet* X que também estão presentes no *ItemSet* resultante da interseção de X com Y, traduzida matematicamente para  $Conf(r) = \frac{|X \cap Y|}{|X|}$ .

Para o entendimento de regras de associação, também é importante definir *ItemSet* frequente e *ItemSet* raro como sendo, respectivamente, aquele cujo suporte é maior que suporte mínimo escolhido previamente e menor que esse suporte mínimo definido. Assim, uma regra r é dita frequente ou rara quando o *ItemSet* resultante de  $X \cup Y$  é qualificado como frequente ou raro, respectivamente.

### 2.2.2 Propriedades dos *ItemSets*

O trabalho desenvolvido por Szathmary, Valtchev e Napoli(4) indica algumas definições relacionadas a *ItemSets* que são de extrema importância para entender as suas diversas representações e saber quais delas são as mais úteis para cada situação.

Esse artigo define o conceito de *ItemSet* fechado no qual o *ItemSet* não possui nenhum superconjunto próprio que possua o mesmo suporte. Um outro conceito relacionado e que é de alguma forma oposto ao anterior é o de *ItemSet* gerador, onde o conjunto não possui nenhum subconjunto próprio com o mesmo suporte.

Além disso, para a mineração de regras de associação, é muito importante a definição de *ItemSet* frequente maximal que engloba todos os *ItemSets* frequentes cujos supersets próprios não são frequentes. De maneira análoga, um *ItemSet* raro minimal é aquele cujos subconjuntos próprios não são raros, e então, a partir desse último conceito pode-se definir o gerador raro minimal sendo aquele o qual todos os subconjuntos próprios não são raros. Como consequência, todo *ItemSet* minimal é também um gerador.

Agora, suponha que os itens dentro de um *ItemSet* estejam lexicograficamente ordenados, como foi pontuado por (23) dois *ItemSets* estão na mesma classe de equivalência se eles compartilham o mesmo prefixo de tamanho k, onde o tamanho é o número de itens que compõem o prefixo. Por exemplo, seja a seguinte classe de equivalência com prefixo de tamanho 1: {A}, {B}, {C}, {D}, os *ItemSets* {A, D}, {A, B} e {A, E} se encontram na mesma classe de equivalência [A] pois compartilham o mesmo prefixo de tamanho 1.

### 2.2.3 Regras de Associação de Multirrelação em Grafos Direcionados

Como foi mencionado anteriormente, as triplas do formato RDF podem ser vistas como um grafo direcionado, o que torna natural que uma técnica de mineração de regras em *databases* cujos dados se encontrem no formato RDF explore esse tipo de representação. Para que isso possa ser feito, porém, se faz necessário adaptar e introduzir alguns conceitos relacionados à mineração de regras de associação para que eles se encaixem ao novo paradigma.

O artigo (2) introduziu uma nova classe de regras de associação, as chamadas regras de associação de multirrelação. A fim de formalizar os conceitos relacionados a essas regras, algumas definições serão apresentadas a seguir. Elas foram retiradas do trabalho de (5) sendo adaptadas para melhor se encaixar nos objetivos desse estudo, com os seus exemplos apresentados fazendo referência à Figura 4.

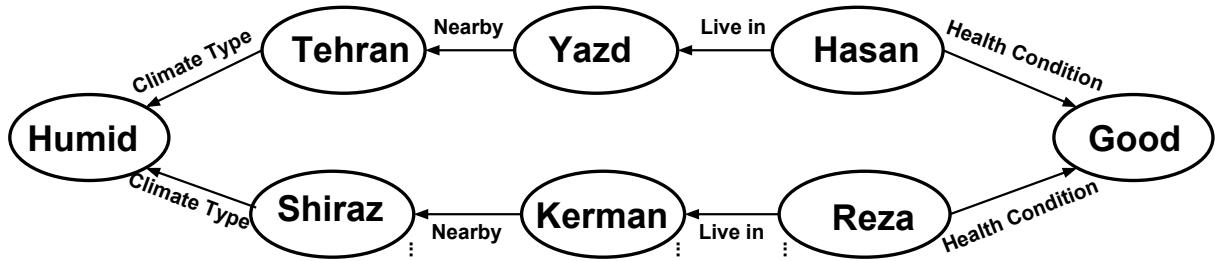


Figura 4 – Exemplo de um grafo direcionado (com arestas rotuladas). Subgrafo extraído de (2).

**Definição 2.2.1.** Um caminho  $C(x, y)$  é um conjunto ordenado de relações  $p^1, \dots, p^k$  que, quando aplicadas ao recurso  $x$ , levam ao recurso  $y$ .

$$C(Hasan, Humid) = Hasan \xrightarrow{\text{Live\_in}} Yazd \xrightarrow{\text{Near\_by}} Tehran \xrightarrow{\text{Climate\_Type}} Humid$$

**Definição 2.2.2.** Uma *ItemChain*  $\mathcal{IC}_{y,p}$  de um grafo direcionado  $G$  é o conjunto de caminhos  $\{C(x_1, y), C(x_2, y), \dots\}$  que, partindo de diferentes recursos, chegam ao recurso  $y$  através da sequência de relações  $p = p^1, \dots, p^k$ , onde  $y$  é chamado de recurso, ou entidade, *endpoint*. Ou seja,  $\mathcal{IC}_{y,p} = \{\{C(x_1, y), C(x_2, y), \dots\} \mid p = p^1, \dots, p^k$  aplicadas a  $\{x_1, x_2, \dots\}$  levam ao recurso  $y\}$ . Os caminhos abaixo, por exemplo, fazem parte da *ItemChain*  $\mathcal{C}_{Humid, (Live\_in, Near\_by, Climate\_type)}$ , sendo *Humid* a entidade *endpoint*:

$$Reza \xrightarrow{\text{Live\_in}} Kerman \xrightarrow{\text{Near\_by}} Shiraz \xrightarrow{\text{Climate\_Type}} Humid$$

$$Hasan \xrightarrow{\text{Live\_in}} Yazd \xrightarrow{\text{Near\_by}} Tehran \xrightarrow{\text{Climate\_Type}} Humid$$

Para que seja possível utilizar os teoremas envolvendo *ItemSets* em *ItemChains* e usufruir da gama teórica já existente para o primeiro na comunidade científica, será preciso

relacionar uma *ItemChain* com um *ItemSet*. Assim, usaremos a adaptação da estrutura de um item dada por Ramezani, Saraee e Nematbakhsh(2), onde, um item é definido como uma entidade, caso o *dataset* contenha somente uma relação entre entidades, ou é definido como um par (entidade, relação) se houver mais de uma relação no *dataset*, como mostra a Figura 5.

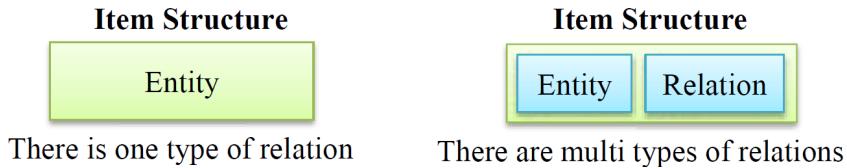


Figura 5 – Estrutura de um Item, extraída de (2).

Como descrito na definição 2, uma *ItemChain* é ilustrada na Figura 6. Assim, de acordo com o conceito de item podemos expressar a *ItemChain* como um conjunto formado pelos pares: (Entidade  $x_1$ , Relação  $p_n$ ), (Entidade  $x_2$ , Relação  $p_n$ ), .., (Entidade  $x_n$ , Relação  $p_n$ ), .., (Entidade A, Relação  $p_1$ ), .. e (Entidade B, Relação  $p_1$ ).

Analizando os pares acima, é possível perceber que cada par é um item, fazendo com que a *ItemChain* genérica apresentado na Figura 6 seja formada por um conjunto de itens, caracterizando um *ItemSet*. Assim, podemos dizer que uma *ItemChain* é também um *ItemSet*.

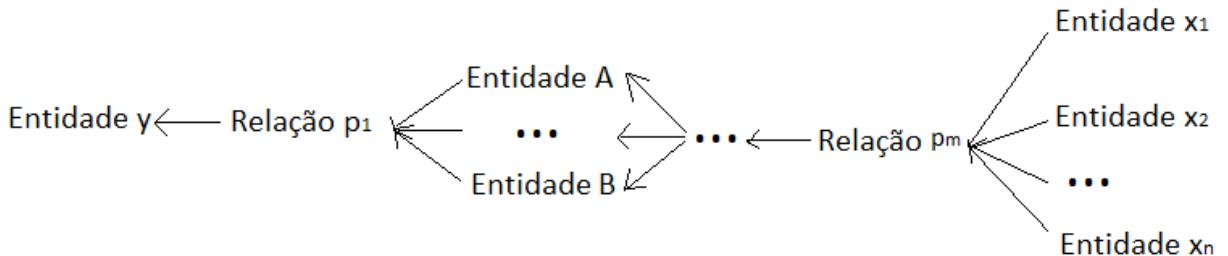


Figura 6 – Estrutura de uma *ItemChain*

**Definição 2.2.3.** Definiremos  $\mathcal{I}(\mathcal{IC}_{y,p})$  como o conjunto de recursos  $\{x_1, x_2, \dots\}$  que, quando aplicados a sequência de relações  $p = p^1, \dots, p^k$  chegam no recurso *endpoint*  $y$ . Pelo exemplo da Definição 2.2.2,  $\mathcal{I}(\{\mathcal{IC}_{Humid,(Live\_in,Near\_by,Climate\_type)}\}) = \{Reza, Hasan\}$ .

**Definição 2.2.4.** Uma *LargeItemChain*  $\mathcal{LIC} = \{\mathcal{IC}_{y_1,p_1}, \mathcal{IC}_{y_2,p_2}, \dots\}$  é um conjunto de *ItemChain*, onde  $\mathcal{IC}_{y_1,p_1}, \mathcal{IC}_{y_2,p_2}, \dots$  são *ItemChains*.

**Definição 2.2.5.** Uma regra de associação de multirrelação  $R$  é definida como  $R : \mathcal{IC}_{y_1,s_1}, \mathcal{IC}_{y_2,s_2}, \dots \rightarrow \mathcal{IC}_{z_1,r_1}, \mathcal{IC}_{z_2,r_2}, \dots$ , onde  $\mathcal{IC}_{y_1,s_1}, \mathcal{IC}_{y_2,s_2}, \mathcal{IC}_{z_1,r_1}, \mathcal{IC}_{z_2,r_2}$  são *ItemChains*, sendo  $\mathcal{IC}_{y_1,s_1}, \mathcal{IC}_{y_2,s_2}, \dots$  o antecedente da regra e  $\mathcal{IC}_{z_1,r_1}, \mathcal{IC}_{z_2,r_2}, \dots$  o consequente da regra, com  $\{\mathcal{IC}_{y_1,s_1}, \mathcal{IC}_{y_2,s_2}, \dots\} \cap \{\mathcal{IC}_{z_1,r_1}, \mathcal{IC}_{z_2,r_2}, \dots\} = \emptyset$ . Usando o exemplo anterior, podemos

definir a regra:

$R : \text{Live\_In}(\text{Near\_By}(\text{Climate\_Type}(\text{Humid}))) \rightarrow \text{Health\_Condition}(\text{Good})$ . Sendo  $R$  um exemplo de regra de associação de multirrelação que indica que se um indivíduo mora numa cidade perto de outra cidade que tem clima úmido, esse indivíduo tem condição de saúde boa.

**Definição 2.2.6.** Uma regra de associação de multirrelação  $R$ , como definida em 2.2.5, é dita frequente (resp. rara) se e somente se  $\text{Supp}(R) \geq \text{minSupp}$  (resp.  $\text{Supp}(R) < \text{minSupp}$ ), onde  $\text{Supp}(R) = |\mathcal{I}(\mathcal{IC}_{y_1,s_1}) \cap \mathcal{I}(\mathcal{IC}_{y_2,s_2}) \dots \cap \mathcal{I}(\mathcal{IC}_{z_1,r_1}) \cap \mathcal{I}(\mathcal{IC}_{z_2,r_2}) \dots| / |V|$ , com  $V$  sendo o número total de nós (recursos) do grafo direcionado analisado.

**Definição 2.2.7.** Uma regra de associação de multirrelação frequente (resp. rara)  $R$ , como definida em 2.2.6, é dita válida se e somente se  $\text{Conf}(R) \geq \text{minConf}$  (resp.  $\text{Conf}(R) = 1$ ), onde  $\text{Conf}(R) = |\mathcal{I}(\mathcal{IC}_{y_1,s_1}) \cap \mathcal{I}(\mathcal{IC}_{y_2,s_2}) \dots \cap \mathcal{I}(\mathcal{IC}_{z_1,r_1}) \cap \mathcal{I}(\mathcal{IC}_{z_2,r_2}) \dots| / |\mathcal{I}(\mathcal{IC}_{y_1,s_1}) \cap \mathcal{I}(\mathcal{IC}_{y_2,s_2}) \dots|$ .

**Definição 2.2.8.** Definiremos  $\mathcal{S}(R)$  como o conjunto de recursos que dão suporte à regra de associação de multirrelação  $R$  definida em 2.2.5. Assim,  $\mathcal{S}(R) = \mathcal{I}(\mathcal{IC}_{y_1,s_1}) \cap \mathcal{I}(\mathcal{IC}_{y_2,s_2}) \dots \cap \mathcal{I}(\mathcal{IC}_{z_1,r_1}) \cap \mathcal{I}(\mathcal{IC}_{z_2,r_2}) \dots$

**Definição 2.2.9.** De forma semelhante a definição 2.2.8, definiremos  $\mathcal{S}(R^*)$  como o conjunto de recursos que dão suporte a um conjunto de regras  $R^* = \{R_1, R_2, \dots, R_n\}$  como sendo  $\mathcal{S}(R^*) = \bigcup_{i=1}^n \mathcal{S}(R_i)$ . Assim,  $\mathcal{S}(R^*)$  representa o conjunto de todos os recursos suporte a todas as regras mineradas presentes no conjunto  $R^*$ .

### 3 TRABALHOS RELACIONADOS

A seguir, serão descritos os trabalhos que estão de alguma forma relacionados ao presente projeto. O primeiro deles, MRAR+, pode ser visto como uma motivação para a existência deste projeto. Nele, foi apresentada uma abordagem de descoberta de associações entre *datasets* da Web Semântica, porém, foi com a sua recorrente utilização em diferentes *datasets* que foi encontrada uma deficiência com relação as regras raras e se viu a necessidade de expandir o escopo desse trabalho visando abranger melhor essa situação específica.

Nos trabalhos seguintes foram apresentados diferentes algoritmos para mineração de regras raras em banco de dados. A primeira abordagem está focada na definição de múltiplos suportes mínimos, já a segunda visa transpor a barreira imposta pela definição de um suporte mínimo, porém, ambas se baseiam em bancos de dados relacionais não tratando da estrutura em grafo direcionado presente no algoritmo MRAR, (2), e necessária para se trabalhar com bancos de dados presentes na Web de Dados.

Como resultado do estudo desses artigos, uma solução baseada em grafos e que busca transpor as barreiras do suporte mínimo foi proposta. Essa nova abordagem, que será detalhada mais adiante no 4 manteve a essência do MRAR+ na descoberta de associações na Web Semântica e a combinou com a visão de (4) para a mineração de regras raras.

#### 3.1 MRAR+

A Web de Dados, por ser rica em informações, viabiliza que ferramentas de extração de conhecimento sejam utilizados em seus *datasets* para descobrir conhecimentos acerca dos mesmos. Além disso, uso dessas ferramentas em conjunto com processos de enriquecimento de recursos em seus *datasets* faz com que seja possível fornecer conhecimento ao usuário sobre esses *datasets* ou, até mesmo, adicionar mais informações a eles no âmbito da Web Semântica.

Nesse sentido, foram desenvolvidos trabalhos ao redor do mundo utilizando os mais diferentes métodos de extração de conhecimento, como o NER (Reconhecimento de Entidades Nomeadas) de (24), e a mineração de regras de (3). Ao contrário do NER, que depende do *schema* do *dataset* e de recursos literais presentes, a mineração de regras é aplicável em uma gama maior de *datasets* pois depende somente dos *schemas* dos mesmos.

O MRAR+, de (3), utiliza o método de mineração de regras frequentes entre *datasets* da Web Semântica. A abordagem dessa ferramenta, apesar de ter como motivação a dificuldade de enriquecer um *dataset* com muitas ligações externas, devido ao grande

esforço computacional envolvido, sendo necessário priorizar recursos, ela também pode ser utilizada para fornecer conhecimento na esfera da Web Semântica, possibilitando ao usuário adicionar mais informações ao conjunto de dados.

O algoritmo usado no MRAR+ para a mineração de regras de associação de multirrelação em grafos direcionados é o MRAR que possui o fluxo de trabalho apresentado na Figura 7 que foi retirado dele. Esse algoritmo, que é baseado no Apriori, foi desenvolvido para encontrar caminhos frequentes no grafo direcionado do *dataset*, ou seja, caminhos que acontecem com uma frequência maior do que a fornecida como entrada pelo usuário, gerando as regras de associação de multirrelação relacionadas a esses caminhos.

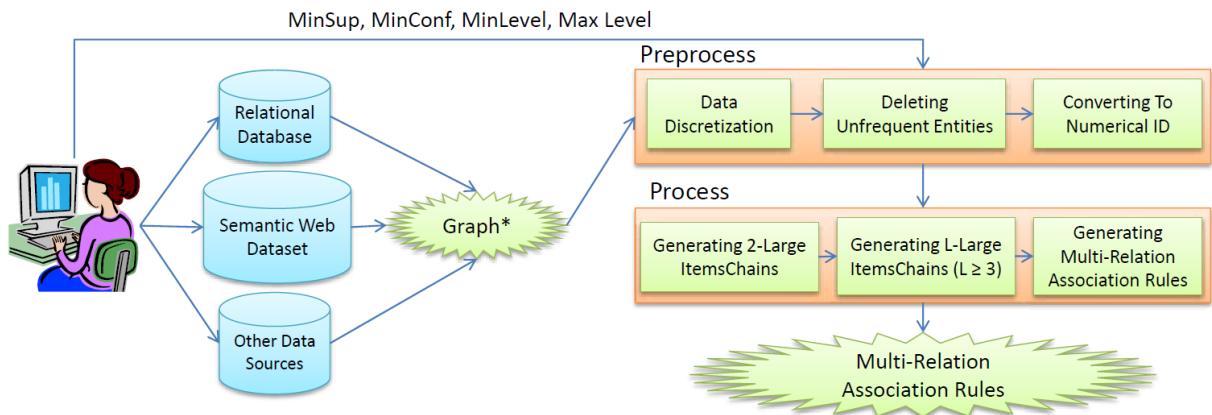


Figura 7 – Processo de Trabalho do MRAR, extraída de (2)

Assim, o MRAR+ foi planejado para ser executado em 6 passos, conforme a Figura 8, sendo usado o algoritmo MRAR somente nos passos 1 e 4. Como entrada, o usuário fornece algumas informações importantes para o funcionamento do MRAR, como o suporte mínimo, a confiança mínima, o nível mínimo e o máximo das regras a serem geradas, além do *dataset* a ser executado. Com isso, o primeiro passo é executado, gerando as regras de associação de multirrelação frequentes que existem no conjunto de dados analisados.

Após isso, nos passos 2 e 3, o *dataset* é ampliado com as informações encontradas sobre os recursos que geraram as regras acima e que possuem ligações semânticas para *datasets* externos. Assim, é o momento de realizar a mineração de regras novamente, no passo 4, para então no passo 5 verificar se houve a geração de uma regra nova devido a ampliação do *dataset*. E, por fim, no passo 6 é possível realizar o ajuste do suporte mínimo utilizado na mineração do *dataset* enriquecido, voltando para o passo 4.

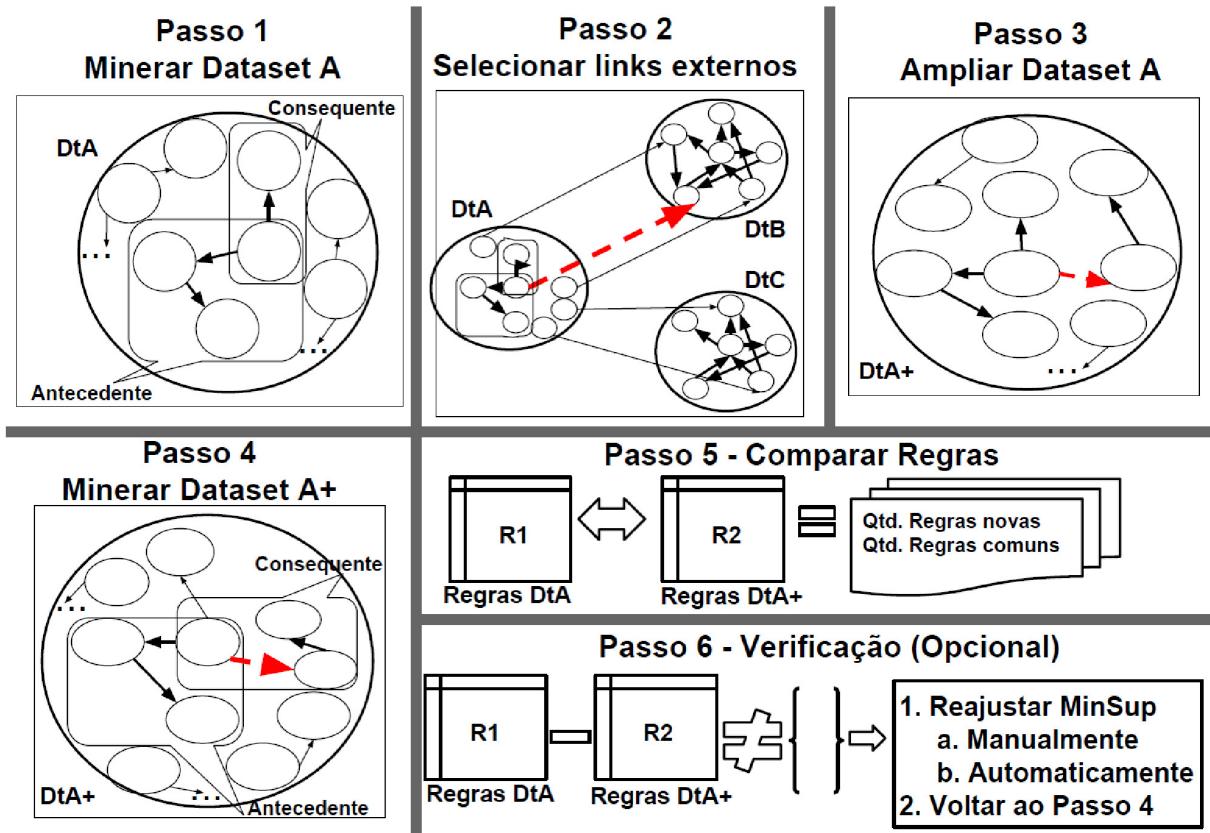


Figura 8 – Visão Geral do MRAR+, retirada de (3)

### 3.2 Mineração de Regras de Associação com Múltiplos Suportes Mínimos

Essa abordagem utilizada por (22) é baseada na análise do comportamento de compras de clientes de um mercado. Ela tem como objetivo descrever como os itens comprados por um cliente estão relacionados entre si. Por exemplo, a regra de associação: *queijo → cerveja* [Supp = 10%, Conf = 80%] significa que 10% dos consumidores compram cerveja e queijo juntos, e 80% das vezes aqueles que compram queijo também compram cerveja. O artigo define que dado um conjunto de transações, o problema de minerar regras de associação significa achar todas as regras de associação que possuem confiança e suporte acima dos mínimos especificados pelo usuário.

O que torna o processo de mineração efetivamente viável é a configuração do suporte mínimo, pois ele restringe o espaço de procura e o número de regras geradas. O problema dessa abordagem é que ela assume que todos os itens do *dataset* são da mesma natureza e aparecem com a mesma frequência no *dataset*, o que não é razoável em um banco de dados real, onde vários itens aparecem em várias transações e outros raramente aparecem. E isso dá margem para dois problemas serem observados quando a frequência

dos itens varia muito dentro de um dado *dataset*:

- Se um suporte mínimo muito alto for estabelecido, não será possível gerar regras que envolvam itens raros, ou seja, não frequentes.
- Para gerar regras que envolvam tanto itens frequentes quanto itens raros, é preciso definir um suporte mínimo muito baixo. Com isso, uma quantidade muito grande de regras será gerada e muitas delas não terão nenhum significado real no banco de dados.

O artigo cita algumas abordagens diferentes que vinham sendo testadas para contornar os problemas acima. Ele menciona que, em geral, o *dataset* era dividido de forma a agrupar itens com frequências parecidas e então a mineração era realizada dentro desses grupos. Uma outra possibilidade era agrupar itens raros relacionados em um único item abstrato que seria mais frequente e então realizar a mineração.

A primeira abordagem citada torna difícil gerar regras entre itens de grupos diferentes, e a segunda torna inviável achar regras que envolvam itens individuais do item abstrato criado. A abordagem proposta pelo artigo possibilita que o usuário escolha diferentes suportes mínimos para a mineração de regras, de forma que eles se adéquem às variações de frequência entre os itens, mais especificamente, ele possibilita que o usuário determine diferentes suportes mínimos para cada um dos itens do conjunto de dados.

O modelo proposto no artigo muda a definição de suporte mínimo e passa a usar o conceito de suporte mínimo de item(MIS) para os itens que aparecem na regra, ou seja, cada item no *dataset* tem um suporte mínimo de item especificado pelo usuário. Dessa maneira, diferentes regras possuem diferentes suportes mínimos que precisam ser satisfeitos.

Dessa maneira, o suporte mínimo de uma regra passa a ser definido como o menor suporte de item (MIS) entre os itens que fazem parte da regra. Seja  $(MIS)_j$  o mínimo suporte de item associado ao item j, a regra R dada por  $a_1, a_2, \dots, a_k \rightarrow a_{k+1}, \dots, a_r$  satisfaz o suporte mínimo se o suporte da regra de fato é maior ou igual a:

$$\min(MIS(a_1), MIS(a_2), \dots, MIS(a_r))$$

Os algoritmos de mineração de regras de associação possuem tipicamente dois passos:

- Achar todos os *ItemSets* largos
- Gerar as regras de associação utilizando esses *ItemSets* largos

A maior parte das pesquisas relacionadas à mineração de dados está focada no primeiro passo que representa um problema que é conhecido por ser computacionalmente custoso, e a exploração da propriedade de fechamento do suporte é responsável por podar o algoritmo de mineração.

O algoritmo proposto nesse trabalho é uma generalização do algoritmo Apriori chamado MSApriori, onde o algoritmo se reduz ao Apriori propriamente dito caso seja admitido apenas um valor de MIS. A procura por *ItemSets* largos é baseada na busca em níveis através de múltiplas passagens pelos dados. Primeiramente, ele conta o suporte individual de cada item e determina se esses itens são largos ou não. Na próxima passagem pelos dados, ele alimenta o algoritmo com o conjunto de itens largos encontrados no passo anterior e os usa para gerar novos *ItemSets* largos possíveis. O suporte verdadeiro desses *ItemSets* candidatos serão computados conforme a passagem pelos dados. No final da passagem, ele determina quais dos candidatos são de fato largos.

### 3.3 Encontrando *ItemSets* Raros e as suas Regras de Associação Raras

Esse artigo, feito por (4), foi motivado por características específicas de banco de dados contendo dados médicos. No caso, o estudo de caso é composto de regras e padrões atípicos pertencentes à um banco de dados biomédicos francês, o STANISLAS que armazena o histórico médico de uma série de famílias consideradas saudáveis da França.

O mais interessante nesse banco de dados é a possibilidade de utilizá-lo para observar características e relações que atingem apenas um pequeno número de indivíduos, essas características seriam atípicas no banco e, portanto, consideradas regras raras.

Dado, por exemplo, um banco de dados que contém o histórico clínico de várias pessoas junto com medicações que elas tomaram e possíveis efeitos adversos que elas tiveram, minerar *ItemSets* relevantes poderia gerar uma associação formal entre as medicações e os efeitos adversos à sua utilização, ou seja, seria possível determinar os efeitos adversos de certos remédios e então usar isso como base para a utilização desses medicamentos em outros tratamentos. Para que esses padrões possam ser detectados, todos os efeitos benignos teriam que ser filtrados primeiro para então focar naqueles menos esperados.

O trabalho defende que abaixar o suporte mínimo para achar regras que envolvam itens não frequentes seja, na verdade, uma abordagem ingênuo do problema de minerar itens raros. Essa medida teria um terrível impacto na performance da mineração de regras e tornaria difícil identificar regras válidas e importantes no meio de tantas que seriam geradas.

Uma outra abordagem que foi estudada durante esse trabalho foi um algoritmo

chamado RSAA (*Relative Support Apriori Algorithm*) que se baseia na especificação de um suporte mínimo relacionado a cada item. As saídas do algoritmo são todos os *ItemSets* e as regras cujo suporte é maior que pelo menos um dos suportes dos itens membros. O MSApriori é uma abordagem semelhante, porém um pouco mais refinada já que modula o suporte de uma regra apenas a partir dos itens que são membros dela.

A abordagem adotada e que será implementada por este artigo é mais radical que as anteriores, pois foca diretamente nos padrões não frequentes que, então, se tornam o objetivo principal da mineração, ou seja, a mineração passa a ser sobre itens raros, ou seja, não frequentes.

Dois artigos se destacam por adotar uma posição semelhante, o AprioriInverse e o *Mining Interesting Imperfectly Sporadic Rules* (MIISR). Contudo, um problema relacionado à ambos é o fato de que eles mineram *ItemSets* envolvendo apenas os itens raros e não aquelas que relacionam os itens raros e aqueles que são frequentes.

A abordagem adotada neste trabalho visa não apenas os *ItemSets* perfeitamente raros, buscando a mineração de regras raras a partir dos *ItemSets* raros encontrados com a mistura de itens raros e frequentes. De maneira semelhante aos *ItemSets* frequentes, os raros podem ser transformados em regra a partir da sua divisão em premissa e conclusão. O resultado será necessariamente raro, mas a confiança pode variar de forma a ser alta o suficiente para ser relevante e considerada uma regularidade no *dataset*.

Contudo, os algoritmos para a mineração de *ItemSets* frequentes são inadequados para extrair regras raras de associação. Eles utilizam o conceito de suporte mínimo que restringe as regras geradas e, portanto, precisam ser alterados.

Para gerar todas as regras raras, é necessário, primeiramente, gerar todos os *ItemSets* raros. Em tese, quando todos os *ItemSets* raros estiverem disponíveis, será possível gerar todas as regras raras válidas. Porém, achar todos esses *ItemSets* é caro em termos de memória devido a quantidade enorme de dados que seria obtido. Além disso, a quantidade de regras geradas a partir de todos esses itens raros seria muito maior, além de muitas serem redundantes e não acrescentarem nenhuma informação relevante.

A representação usada aqui para os *ItemSets* será baseada nos conceitos geradores e *ItemSets* fechados. Além disso, o conjunto de regras de associação minimais e não redundantes serão de extrema importância por ser uma forma informativa e sem perda de representação de todas as regras válidas e frequentes de associação, e por isso, será utilizado um conjunto análogo para representar as regras de *ItemSets* raros minimais.

O nome da abordagem utilizada de chama "*Breaking the barrier*" (Quebrando a Barreira) devido à tentativa de transpor à barreira do suporte mínimo definido e conseguir gerar regras que estejam abaixo dessa barreira, ou seja, regras raras, porém que contenham uma alta confiança associada. Pode-se indicar três passos que devem ser seguidos para a

confecção do algoritmo:

- Computar o conjunto de *ItemSets* raros minimais utilizando o algoritmo AprioriRare. O Apriori original, como mencionado antes, é utilizado para encontrar *ItemSets* frequentes, porém, como efeito colateral, acaba por gerar concomitantemente os *ItemSets* raros minimais (mRIs).
- Achar o fechamento desse *ItemSets* raros minimais e utilizá-los para encontrar os conjuntos de equivalência desses *ItemSets*.
- Explorar os conjuntos raros de equivalência para gerar as regras de associação raras de uma maneira semelhante ao método utilizado para achar regras de associação minimais não-redundantes. Essas regras serão chamadas de mRGs.

No trabalho, o tipo de mRG estudada é a exata, onde é observada a seguinte característica:

$$\begin{aligned}
 & P_1 \subset P_2 \\
 & P_1 \text{ é uma mRG} \\
 & P_1 \cup (P_2 \setminus P_1) \text{ é um } ItemSet \text{ raro fechado} \\
 & conf(r) = 1,0
 \end{aligned}$$

$r : P_1 \Rightarrow P_2 \setminus P_1$ , onde

Essas regras serão, então, regras raras de associação onde o antecedente ( $P_1$ ) é raro e o consequente ( $P_2 \setminus P_1$ ) é raro ou frequente. E  $P_2$  e  $P_1$  estão na mesma classe de equivalência, como pode ser visto na Figura 9.

Como o gerador é um subconjunto minimal de seu fechamento com o mesmo suporte, as regras geradas permitem que máxima informação seja deduzida com as hipóteses mínimas.

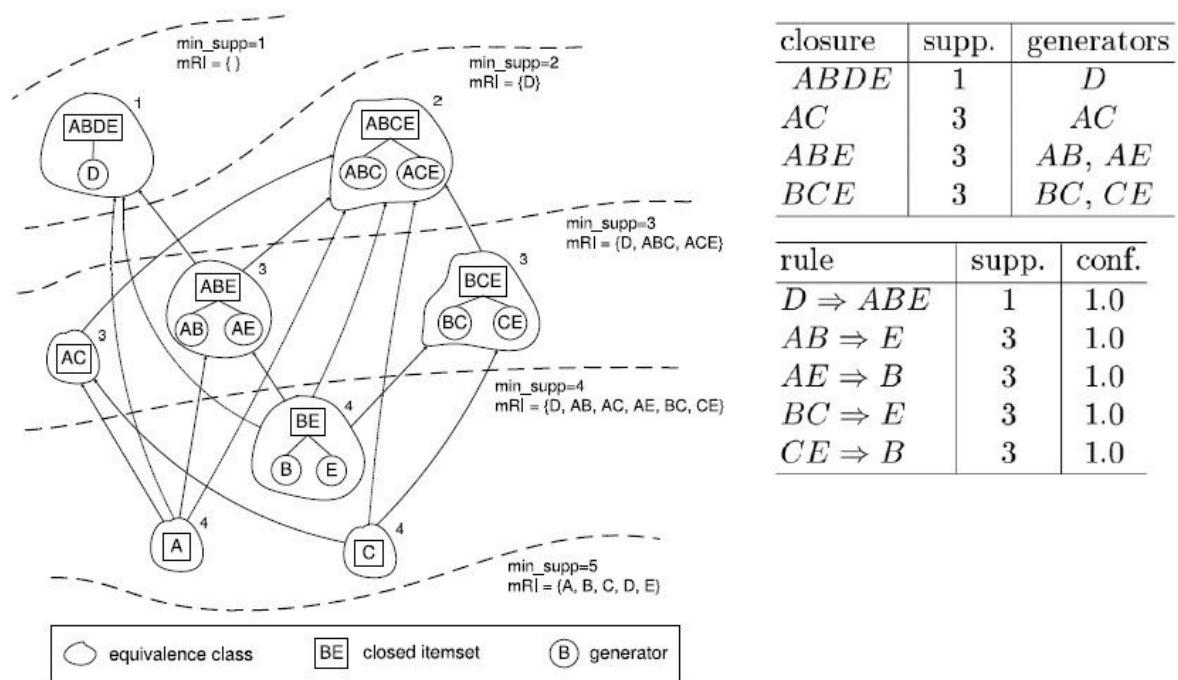


Figura 9 – Esquerda: Classes de equivalência raras de um *dataset*  $D$  com diferentes suportes mínimos; Direita Superior: Classes de equivalência raras de um *dataset*  $D$  com suporte mínimo igual a 4; Centro da Direita: regras mRG exatas em  $D$  com suporte mínimo igual a 4. Recorte de uma Figura extraída de (4)

## 4 MONET: UMA NOVA ABORDAGEM PARA A DESCOBERTA DE ASSOCIAÇÕES ENTRE *DATASETS* DA WEB DE DADOS

A enorme quantidade de *datasets* interligados presente na Web Semântica representa um grande número de informações, e faz com que seja possível usá-la a favor de um *dataset* alvo especificado através da extração de conhecimento. A mineração de regras é um método de extração de conhecimento que pode explorar essa grande quantidade de informações disponíveis ao buscar padrões nos relacionamentos entre os conjuntos de dados. Contudo, a maior parte dos algoritmos de mineração de regras foca somente em regras que representam caminhos frequentes no *dataset*, esquecendo dos padrões sempre verdadeiros, mas pouco frequentes e que também podem ser muito importantes no processo de extração de conhecimento.

Como foi visto na seção 3.1, a abordagem da ferramenta MRAR+ considera somente a mineração de regras frequentes para encontrar associações entre *datasets* da Web Semântica, não possuindo um foco específico para as regras raras que também são úteis ao seu objetivo de descobrir associações na Web Semântica. Por outro lado, os trabalhos apresentados nas seções 3.2 e 3.3, ao estarem limitadas a bancos de dados relacionais, não consideraram a estrutura em grafo direcionado em seus algoritmos propostos para a mineração de regras raras.

O presente trabalho pretende preencher essa lacuna, combinando a abordagem do MRAR+ com o algoritmo de mineração de regras raras de (4), a fim de se criar uma abordagem mais completa no âmbito de descoberta de associações entre as bases de conhecimento existentes na Web de Dados ao utilizado tanto as regras frequentes quanto raras. E assim surgiu o MONET - *Mining fOr iNterlinking wEb daTases*, que combina o MRAR+ de Oliveira et al.(5) e o MRARE, que foi desenvolvido no presente trabalho e que será devidamente apresentado e descrito nas próximas seções.

A escolha do algoritmo de mineração de regras raras proposto por (4) e não do proposto por (22) se justifica pois o segundo necessita de uma análise prévia do banco de dados, sendo necessário definir, para o seu correto funcionamento, um suporte mínimo específico para cada item desse conjunto de dados, o que demandaria muito tempo e esforço para trabalhar com banco de dados que não tenham sido inicialmente concebidos voltados para essa abordagem. Já o algoritmo proposto por (4) não requer qualquer alteração por parte do usuário no *dataset*.

## 4.1 Visão Geral

A Figura 10 apresenta a visão geral da abordagem proposta no presente trabalho, denominada MONET (*Mining fOr iNterlinking wEb daTases*). Ela tem como intuito usar as informações existentes entre um *dataset* da Web Semântica e um *dataset* alvo especificado pelo usuário e que podem ser obtidas através da mineração de regras tanto frequentes quanto raras. Além disso, é importante destacar que o algoritmo utilizado para a mineração de regras frequentes foi o MRAR. Já o MRARE, que foi desenvolvido ao longo deste trabalho e será detalhado na próxima seção, foi usado para a mineração de regras raras.

A abordagem proposta irá comparar as regras frequentes e/ou raras geradas antes e depois da ampliação do *dataset* com as informações externas e determinar as regras frequentes e/ou raras que foram originadas com o processo de ampliação do *dataset* e que não existiam anteriormente.

Baseada no funcionamento do MRAR+ do trabalho (3), a nova abordagem terá como entrada um *dataset* contendo ligações externas com outras bases de conhecimento presentes na Web Semântica. Após a leitura dos dados desse *dataset*, junto com alguns parâmetros de entrada necessários para a realização da mineração de regras, será possível minerar regras de associação de multirrelação frequentes e/ou raras no *dataset* em questão.

Com essas regras de associação de multirrelação geradas, podem ser obtidas informações acerca de quais dos recursos desse *dataset* alvo foram os principais responsáveis pelas regras mineradas. Assim, utilizando os meios que a Web Semântica fornece, será possível realizar uma consulta SPARQL sobre esses mesmos recursos, porém em alguma outra base de conhecimento presente na Web de Dados através de propriedades do tipo *owl:sameAs*. Por fim, o resultado dessa busca, que conterá informações sobre esses recursos, será incorporado ao *dataset* alvo (que o usuário especificou como entrada).

Além disso, uma opção permitida ao usuário é visualizar quais recursos serão consultados no *database* externo antes de se realizar a consulta. Com isso, será possível finalizar a execução antes de continuar o algoritmo, caso o usuário veja que não serão feitas consultas sobre muitos recursos ou que as consultas serão feitas em cima dos recursos que ele não esperava (uma vez que a consulta SPARQL depende do tipo de recurso).

Prosseguindo com a ampliação do *dataset* alvo através do resultado da busca SPARQL feita, para que seja possível tirar proveito das novas informações existentes nesse *dataset* será preciso executar mais uma vez a mineração de regras. Contudo, essa segunda mineração terá um suporte mínimo diferente no inicialmente definido como entrada, *MinSupp<sub>initial</sub>*, sendo modificado a fim de que seja possível gerar os mesmos *ItemChains* presentes na mineração anterior.

Isso é necessário uma vez que, aumentando o número de nós no *dataset*, o valor

do suporte das *ItemChains* presentes anteriormente diminui, fazendo com que aquelas que possuíam suporte muito próximo ou igual ao mínimo não fossem mais geradas, sendo, portanto, perdidas as regras que haviam sido geradas a partir delas. Sendo assim, o suporte mínimo da segunda mineração,  $MinSupp_{final}$ , será igual a  $(n_{inicial} * MinSupp)/n_{final}$ , onde  $n_{inicial}$  e  $n_{final}$  são os números de nós no *dataset* antes e depois de sua ampliação, respectivamente.

Como saída dessa segunda mineração, teremos um segundo conjunto de regras que podem envolver as informações que foram adicionadas ao *dataset* ampliado. Dessa forma, essas regras podem ser comparadas com o primeiro conjunto de regras a fim de encontrar aquelas que apareceram somente nesse segundo conjunto.

Assim, o produto do MONET será um conjunto de regras que existem entre o *dataset* alvo e a Web de Dados, ou seja, que foram geradas a partir da ampliação do conjunto de dados. Isso porque se entende que as regras que são geradas sobre o *dataset* alvo refletem o conhecimento pertencente à ele, enquanto que as regras que são geradas sobre o *dataset* ampliado refletem tanto o conhecimento pertencente a ele quanto o que pertence à Web Semântica.

Apesar da proposta do MONET ser a entrega desse conjunto de regras ao usuário, é importante que este analise essas regras novas para que seja possível encontrar inferências acerca de novas triplas (informações) ou conhecimento que possa ser adicionado ao seu *dataset*, e que reflete as ligações existentes no mesmo com a Web Semântica através de ligações *owl:sameAs*.

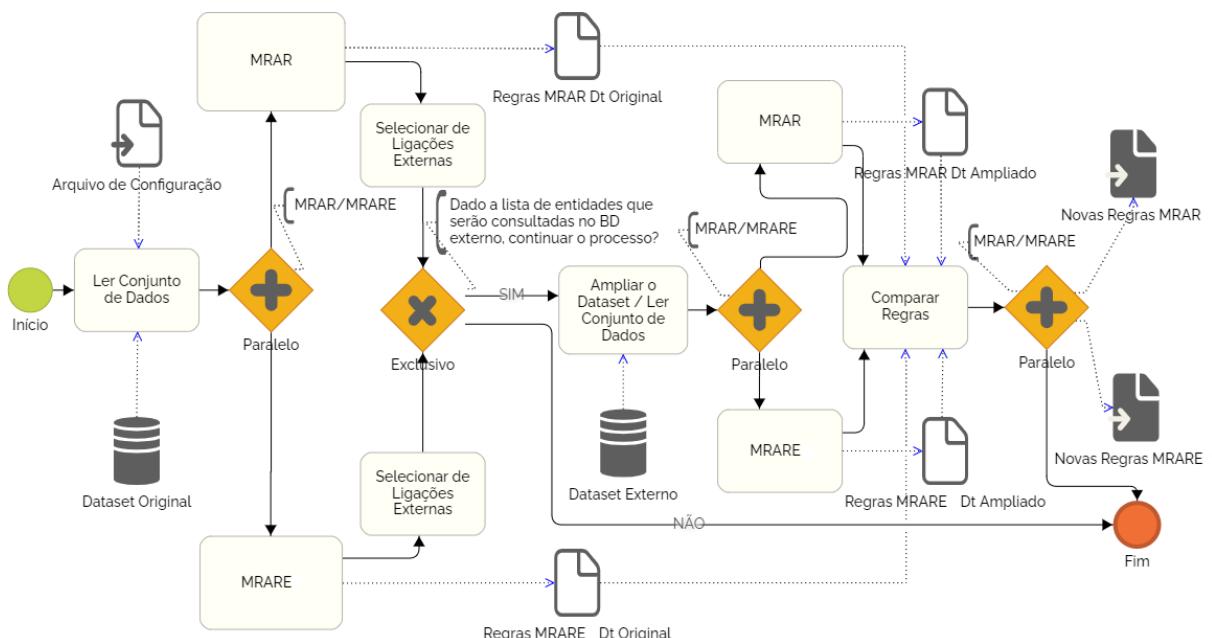


Figura 10 – Visão Geral da Abordagem MONET baseada no *Business Process Model and Notation* (BPMN)

## 4.2 Algoritmo MRARE

O procedimento escolhido para a mineração de regras raras válidas, que está presente no artigo (4), é orientado a *ItemSets*, o que torna a sua abordagem inviável ao se trabalhar com bancos de dados não relacionais como os grafos direcionados presentes na Web de Dados. Assim, o novo algoritmo MRARE adapta esse procedimento para que seja possível de utilizá-lo com *ItemChains*, como proposto nos trabalhos Oliveira et al.(5) e Ramezani, Saraee e Nematbakhsh(2) apresentado em 3.1, e -assim- ser compatível com os *datasets* da Web de Dados.

O pseudocódigo referente ao MRARE, bem como seus principais métodos, será apresentado nesta seção e irá indicar o passo a passo necessário para a mineração de regras raras em *datasets* utilizando o conceito de *ItemChain*. Cada algoritmo apresentado terá suas principais funcionalidades comentadas, a fim de facilitar o entendimento da solução e esclarecer o seu funcionamento.

O pseudocódigo 1 mostra um esboço do algoritmo MRARE propriamente dito com as suas principais funcionalidades, entradas e saídas. No caso, como entradas, teremos o suporte mínimo, o *minLevel* e o *maxLevel*, que como foi dito anteriormente neste trabalho, determinam a quantidade mínima e máxima, respectivamente, de relações entre o *endpoint* e as entidades ligadas a ele. Já como saídas, temos a lista de regras encontradas  $R_L$  e o conjunto de recursos que geraram essas regras  $S_L(R_L)$ .

Primeiramente, será construído um grafo direcionado contendo arestas rotuladas com o nome da relação que essa aresta representa. Então, os dados presentes nesse grafo serão convertidos numa estrutura de mapeamento *EntityInfo*, que alimentará o algoritmo.

O *EntityInfo* se apresentará como um mapa das entidades *endpoint* presentes no *dataset*, como vemos na linha 24 do pseudocódigo 1, sendo uma forma de alimentar o algoritmo sem buscas desnecessárias, o que auxilia no trabalho com *datasets* muito grandes. Assim, o *EntityInfo* será composto por uma chave que será o *endpoint* (objeto) da tripla e como valor associado a essa chave teremos um outro mapa, esse último mapa terá como chave uma relação (predicado) e como valor associado a essa nova chave estará o conjunto de recursos (sujeitos) que estão ligados ao respectivo *endpoint* através da relação mencionada, como é mostrado na Figura 11.

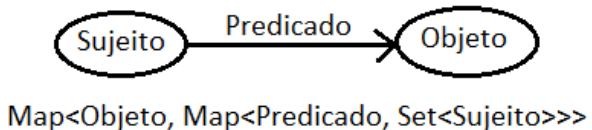


Figura 11 – Estrutura *EntityInfo*

O *EntityInfo* é percorrido a partir das chaves de sua estrutura de mapeamento,

---

**Algorithm 1** MRARE: Mining Multi-Relation Association Rare Rules

---

```

1: Input
2:   DS           ▷ A dataset convertible to a directed graph with labeled edges
3:   MinSupp      ▷ Minimum support value
4:   MinConf      ▷ Minimum confidence value
5:   MinLevel, MaxLevel    ▷ Minimum and maximum number of relations in each
   ItemChain
6: EndInput
7: function MRARE(DS, MinSupp, MinConf, MinLevel, MaxLevel)
8:   Output
9:     AllLICs      ▷ Map of Rare ItemChains about Large ItemChains
10:    Rules         ▷ Multi-Relation Association Rare Rules
11:   EndOutput
12:   Variables
13:     LLFrequentICs[]          ▷ List of Large ItemChains
14:     LLRareICs[]             ▷ List of Large ItemChains
15:     Candidates[]            ▷ Lists that maintain ChainIDs Set
16:     CIS                      ▷ Set of ChainIDs
17:     LIC1, LIC2              ▷ Large ItemChain
18:     List EntityInfo[]        ▷ List of large EntityInfo instances
19:     List FrequentItemChains[] ▷ Frequent list of ItemChains
20:     List RareItemChains[]    ▷ Rare list of ItemChains
21:   EndVariables
22:   Convert input data to a directed graph with labeled edges
23:   Construct EntityInfo from the input graph
24:   for EntityInfo in Map EntityInfo<> do
25:     for Relation in EntityInfo.Relations do
26:       GenerateItemChains(EntityInfo.EndpointEntity, Relation, EntityInfo.Re-
          lationEntities, 1)          ▷ adds ItemChains to List FrequentItemChains and List
          RareItemChains
27:     end for
28:   end for
29:   AllLICs<> = GenerateAllLargeRareItemChains(RareItemChains[], Freque-
          ntItemChains[])
30:   Rules = GenerateRareRules(AllLICs<>)
31:   return Rules, AllLICs<>
32: end function

```

---

que são as entidades *endpoint*, e o método **GenerateItemChains** é chamado para cada relação presente nessas entidades *endpoint*, como podemos ver na linha 24 e na linha 25 do pseudocódigo 2. Assim, essa chamada de função receberá como entrada a entidade *endpoint*, que também é um dos vértices do grafo direcionado por onde o algoritmo inicia a sua busca para a mineração das regras, além dos vértices do grafo que estão conectados a essa entidade através de uma relação em comum, juntamente com essa relação.

Como resultado da execução dessa primeira chamada do método **GenerateItemChains**, será possível gerar as *ItemChains*, sendo o seu funcionamento detalhado no

---

**Algorithm 2** GenerateItemChains: Subrotina do MRARE

---

```

1: Input
2:   EndpointEntity           ▷ A vertex of graph which the algorithm starts search from it
3:   Relations Parameter[]    ▷ Common relations between vertices and the EndpointEntity
4:   Entities Parameter[]     ▷ Vertices connected to EndpointEntity through Relations Parameter
5:   Level                     ▷ Number of relations in ItemChain, initially it is 1
6: EndInput
7: function GENERATEITEMCHAINS(EndpointEntity, Relations Parameter[], Entities Parameter[], Level)
8:   Output
9:
10:  List FrequentItemChains[]          ▷ List of Frequent ItemChains
11:  List RareItemChains[]             ▷ List of Rare ItemChains
12: EndOutput
13: Variables
14:  Entities Var[]                 ▷ List of entities
15:  Relations Var[]                ▷ List of relations
16:  Support                       ▷ Support value of an ItemChain
17:  ChainID                       ▷ ID of the ItemChain
18: EndVariables
19: if (Level >= MinLevel Level <= MaxLevel) then
20:   Support = Entities_Var.Count ÷ Graph.NumberOfVertices
21:   if (Support >= MinSup AND Entities_Var.Count > 1) then:
22:     ChainID = ChainID + 1
23:     List FrequentItemChains.Add(new ItemChain(frequentChainID, Entities Var, Re-
        lations Parameter, EndpointEntity, Support))
24:   else
25:     if (Entities _Var.Count > 1) then:
26:       ChainID = ChainID + 1
27:       List Rare1ItemChains.Add(new ItemChain(rareChainID, Entities Var, Relati-
        ons Parameter, EndpointEntity, Support))
28:     end if
29:   end if
30: end if
31: if (Level < MaxLevel) then:
32:   Relations Var = UnionIncomingEdgesOf(Entities Var)
33:   for all (Relation in Relations Var) do
34:     Entities Var = List of vertices that are connected To EndpointEntity through
        Relations Parameter
35:     GenerateItemChains(EndpointEntity, Relations Parameter U Relation, Entities
        Var, Level + 1)
36:   end for
37: end if
38: end function

```

---

pseudocódigo 2. Primeiramente, veremos se o número de relações presentes associadas a uma chamada do método, *Relations\_Parameter[]*, é maior ou igual a um número mínimo (*minLevel*) de relações e menor ou igual a um número máximo (*maxLevel*). Assim, caso satisfaça a essas condições, então ocorre o cálculo do suporte fazendo a divisão entre o número de entidades presentes na lista de entidades conectadas ao *endpoint* através da lista

de relações *Relations\_Parameter[]*, *Entities\_Parameter[]* (entidades que dão suporte à regra), e o número total de vértices presentes no grafo.

Prosseguindo com o funcionamento da subrotina *GenerateItemChains*, será realizada uma comparação na linha 2 para determinar se o suporte calculado é maior ou igual (resp. menor) ao suporte mínimo definido pelo usuário e se o possível *ItemChain* em questão foi gerada por mais de um recurso. Essa restrição de possuir mais de um recurso gerando o *ItemChain* serve para não gerar uma regra que faz menção a somente um recurso, algo possível de acontecer, uma vez que o suporte dessa *ItemChain* pode ser muito pequeno em regras raras.

Assim, caso essa desigualdade seja satisfeita, então será criada uma nova *ItemChain* utilizando como parâmetros de inicialização a entidade *endpoint*, a lista de entidades conectadas ao *endpoint*, as relações comuns entre essas entidades e esse *endpoint*, o seu suporte calculado e o *ChainID*, um identificador único do *ItemChain*. Por fim, logo após a sua criação, a *ItemChain* será adicionada a lista de *FrequentItemChains* (resp. *RareItemChains*).

Por outro lado, se o número de relações presentes na chamada na função for menor que o número máximo (*maxLevel*) de relações, então é possível chamar novamente a função com mais uma relação. Para isso, haverá uma busca na estrutura *EntityInfo* para encontrar o conjunto das relações que chegam na lista de entidade que estão conectadas ao *endpoint* através das relações comuns, *Entities\_Var[]*.

Após isso, o algoritmo irá iterar dentro desse conjunto de relações encontradas, onde, durante cada iteração, o algoritmo adicionará a lista de relações comuns a respectiva relação a ser iterada. Por fim, será buscado quais são as novas entidades que estão conectadas ao *endpoint* através dessa nova lista de relações comuns, sendo chamado o algoritmo de *GenerateItemChains* para cada uma dessas iterações.

Na próxima etapa, o algoritmo 1 chamará o pseudocódigo 3, *GenerateAllLargeRareItemChains*, que recebe como parâmetros de entrada as listas *RareItemChains* e *FrequentItemChains*, tendo como saída uma estrutura que faz o mapeamento entre a *ItemChain* geradora e o seu closure, que é uma *LargeItemChain*. Nesse método, para cada *ItemChain*  $\mathcal{IC}_1$  na lista de *RareItemChains* será pego uma *ItemChain*  $\mathcal{IC}_2$  da lista de *RareItemChains* ou de *FrequentItemChains* e, então, pegaremos a interseção *LOE* das entidades conectadas com o respectivo *endpoint* entre o primeiro *ItemChain* (raro) e o segundo *ItemChain* (raro ou frequente), ou seja,  $LOE = \mathcal{I}(\mathcal{IC}_1) \cap \mathcal{I}(\mathcal{IC}_2)$ .

Essa interseção será a lista de entidades que aparecem em ambas as *ItemChains*. O suporte então será calculado fazendo a divisão entre o número de entidades presente em *LOE* e o número total de vértices do grafo. Se esse suporte for igual ao do *ItemChain* raro  $\mathcal{IC}_1$ , então quer dizer que ambas *ItemChains* compartilham o mesmo conjunto de entidades

---

**Algorithm 3** GenerateAllLargeRareItemChains

---

```

1: Input
2:   List RareItemChains[]                                ▷ List of rare ItemChains
3:   List FrequentItemChains[]                            ▷ List of frequent ItemChains
4: EndInput
5: function GENERATEALLLARGEITEMCHAINS(List RareItemChains[], List Fre-
   quentItemChains[])
6:   Output
7:     Map AllLICs<ItemChain, List ItemChains[]> ▷ Map of all Large ItemChains
8:   EndOutput
9:   Variables
10:    IC1, IC2                                         ▷ ItemChain
11:    LOE[]                                            ▷ List of Entities
12:   EndVariables
13:   for all (IC1 in List RareItemChains) do
14:     for all (IC2 in RareItemChains,FrequentItemChains) do
15:       LOE = Intersect(IC1.LOE, IC2.LOE)
16:       Support = LOE.Length ÷ Graph.NumberOfVertices
17:       if (Support == IC1.support) then
18:         AllLICs(IC1).Add(IC2)
19:         AllLICs(IC1).Add(IC1)
20:       end if
21:     end for
22:   end for
23:   return AllLICs<>
24: end function

```

---

e, portanto, pertencem ao mesmo *closure*. Assim, um mapeamento será criado em AllLICs, tendo como chave o *ItemChain* raro  $\mathcal{IC}_1$  e como valor dessa chave um *LargeItemChain* contendo ambas as *ItemChains*  $\mathcal{IC}_1$  e  $\mathcal{IC}_2$  selecionadas.

Por último, temos o pseudocódigo 4, GenerateRareRules, que gera as regras raras. Esse algoritmo irá pegar o mapa AllLICs que obtivemos através de GenerateAllLargeRareItemChains, como retorno da função vista no pseudocódigo 3, e então, para cada *ItemChain* geradora  $\mathcal{IC}_{geradora}$  que é também a chave desse mapeamento, será selecionada a *LargeItemChain*  $\mathcal{LIC}$  que é o valor desse mapeamento.

Assim, a *ItemChain* geradora  $\mathcal{IC}_{geradora}$  será eleita o antecedente da regra  $R$  que será minerada e o consequente será a *LargeItemChain*, subtraindo-se a *ItemChain* geradora,  $\mathcal{LIC} - \mathcal{IC}_{geradora}$ . Além disso, como o suporte da *ItemChain* geradora será o mesmo suporte da regra, já que  $I(\mathcal{IC}_{geradora}) = I(\mathcal{LIC}) = I(\mathcal{LIC} - \mathcal{IC}_{geradora})$ , essa regra sempre terá confiança 1.0, ou seja, confiança máxima.

Por fim, a regra será criada usando o suporte calculado  $Supp(R) = I(\mathcal{IC}_{geradora})/|V|$  e o conjunto de entidades que a geraram e que estão ligadas ao *endpoint* da *ItemChain* geradora através da sua lista de relações comuns  $S(R) = I(\mathcal{IC}_{geradora})$ , onde  $|V|$  é o

---

**Algorithm 4** GenerateRareRules

---

```

1: Input
2:   Map AllLICs<>                                ▷ Mapp of all generators with its closures
3: EndInput
4: function GENERATERARERULES(AllLICs<>)
5:   Output
6:     Rules[]                                         ▷ List of Rare Multi-Relation Association Rules
7:   EndOutput
8:   Variables
9:     LIC                                              ▷ Large ItemChain
10:    Antecedent                                     ▷ A rare ItemChain that appears in the rule antecedente part
11:   EndVariables
12:   for all (GIC in AllLICs) do
13:     LIC = AllLICs.GIC                            ▷ Set of all ItemChains in the LargeItemChain value
14:     LIC = LIC / GIC
15:     Antecedent = GIC
16:     Consequent = LIC
17:     Rules.Add(new Rule(Antecedent, Consequent, 1.0, GIC.Support, GIC.LOE))
18:   end for
19:   return Rules
19: end function

```

---

número total de nós no grafo analisado.

### 4.3 MONET Tool: Implementação da abordagem MONET

A abordagem MONET foi implementada utilizando a linguagem de programação JAVA, e como ela utiliza a ferramenta do MRAR+ como base, o primeiro passo dado foi reescrever o algoritmo do MRAR+ já implementado em PHP agora na linguagem JAVA. Com essa mudança, a expectativa é a de que possamos utilizar bibliotecas e métodos já consolidados para manipulação de banco de dados e consultas SQL, e ainda obter uma maior velocidade em termos de execução do código, além de uma maior legibilidade devido à modularização da solução em diferentes classes com seus próprios métodos e atributos.

A ferramenta denominada MONET Tool foi desenvolvida de maneira que os parâmetros iniciais necessários à sua execução fossem lidos diretamente de um arquivo de configuração *config.xml* escrito no formato XML e cuja estrutura é a seguinte:

Listing 4.1 – Estrutura do arquivo de configuração

```

<?xml version="1.0" encoding="UTF-8"?>
<CONFIGURATION>
  <MODULE>
    <MRAR> </MRAR>
    <MRARE> </MRARE>

```

```

<LOD> </LOD>
</MODULE>

<RULE>
    <SUPPORT>  </SUPPORT>
    <CONFIDENCE>  </CONFIDENCE>
    <LEVEL>
        <MIN>  </MIN>
        <MAX>  </MAX>
    </LEVEL>
</RULE>

<INPUT>
    <SOURCE>
        <FILE>  </FILE>
    </SOURCE>
    <TARGET>
        <PREVIEW>  </PREVIEW>
        <ENDPOINT>  </ENDPOINT>
        <PREFIX>
            <LABEL>  </LABEL>
            <NAMESPACE>  </NAMESPACE>
        </PREFIX>
        <PROPERTY>  </PROPERTY>
    </TARGET>
</INPUT>

<OUTPUT>
    <MRAR>
        <NAME>  </NAME>
        <PATH>  </PATH>
    </MRAR>
    <MRARE>
        <NAME>  </NAME>
        <PATH>  </PATH>
    </MRARE>
</OUTPUT>
</CONFIGURATION>
```

Essas *tags* possuem nomes que indicam que tipo de informação está contida dentro dela ou a que elas se referem, e serão lidas pelo código fonte JAVA durante a sua execução. A seguir será especificado o papel de cada uma das *tags* na configuração da ferramenta proposta:

- MODULE: Nessa *tag* estão especificados os módulos que o usuário pode escolher executar

- MRAR - Duas opções para conteúdo dentro dessa *tag*, ON ou OFF. Caso o usuário especifique o ON, o código irá executar o módulo referente à implementação do MRAR em JAVA. Caso contrário, o código não irá executar esse módulo.
  - MRARE - Também existem duas opções para o conteúdo dentro dessa *tag*, ON ou OFF. Se o usuário escolher a opção ON, o programa irá executar o módulo referente ao MRARE implementado em JAVA. Caso contrário, o código não irá executar esse módulo.
  - LOD - Duas opções para informação contida dentro dessa *tag*, ON ou OFF. Se o usuário escolher a opção ON, a ferramenta realizará consultas em *datasets* externos. Caso contrário, será realizada apenas a mineração no *dataset* que será especificado mais a frente nesse mesmo arquivo de configuração
- RULE: Nessa *tag* estão especificados parâmetros relacionados com as regras que serão mineradas
    - SUPPORT - O conteúdo dentro dessa *tag* indica a fronteira de suporte que o usuário deseja para as regras que serão geradas
    - CONFIDENCE - O conteúdo dessa *tag* indica o nível de confiança que o usuário deseja no processo de mineração de regras frequentes mineradas pelo algoritmo MRAR
    - LEVEL - Dentro dessa *tag* existem outras duas MIN e MAX que indicam respectivamente número mínimo e máximo de relações entre o *endpoint* e as entidades das regras.
  - INPUT: Dentro dessa *tag* temos SOURCE e TARGET
    - SOURCE - Indica o caminho absoluto local para o banco de dados fonte que o usuário deseja utilizar
    - TARGET - Nessa *tag* estarão informações como PREVIEW, onde o usuário poderá indicar se deseja ou não ver as entidades que serão buscadas no banco externo antes que o banco fonte seja ampliado; o ENDPOINT, especificando o endereço *endpoint* a partir de onde será realizado a consulta no banco externo; o PREFIX, onde o usuário pode especificar o prefixo utilizado em sua consulta; e o PROPERTY onde será inserida a relação que será buscada no banco de dados externo a ser consultado.
  - OUTPUT - Nessa *tag* estarão especificados o caminho local absoluto para a salvar as regras geradas a partir da execução do módulo MRAR, caso o usuário tenha decidido executá-lo e o caminho local absoluto para salvar as regras geradas a partir da execução do módulo MRARE.

Como já foi dito anteriormente, escrevemos o código em JAVA visando a modularização para facilitar a legibilidade do programa. Além do algoritmo principal que foi exposto através dos pseudocódigos, algumas classes auxiliares foram implementadas para garantir o correto funcionamento das classes principais. Assim, as principais classes implementadas são as seguintes:

- WorkFlow - Na classe WorkFlow temos o controle sobre a sequência de ferramentas que serão executadas, ou seja, nessa classe está especificado o caminho do fluxograma que será seguido nessa execução do programa. Quando um objeto dessa classe é criado, ele é inicializado com o arquivo de inicialização XML que foi mencionado anteriormente. Nesse arquivo, o programa irá encontrar os parâmetros escolhidos pelo usuário para guiar a execução.

Nessa classe também encontramos o método *start* que irá utilizar a informação LOD que já foi mencionada anteriormente e que irá determinar se ocorrerá a ampliação do banco de dados com a busca de ligações em *datasets* externos e então haverá a mineração ou se a mineração ocorrerá apenas no *dataset* especificado inicialmente.

- Parameter - Na classe Parameter, os parâmetros de execução serão efetivamente lidos a partir do arquivo de configuração em XML e armazenados.
- MRAR - Nessa classe está a implementação do pseudocódigo descrito no artigo publicado do MRAR. Como foi dito anteriormente, esse código já havia sido implementado em PHP no MRAR+ de (3) porém, nesta classe encontramos a reescrita em JAVA de todas as suas funcionalidades incluindo a geração de ItemChains e LargeItemChains e a geração de regras, bem como o método que mostra as regras geradas para o usuário bem como as entidades que deram suporte à mineração dessas regras.
- MRARE - Nessa classe está a implementação em JAVA do pseudocódigo que foi mostrado em 4.2.
- SPARQL - Nessa classe está a implementação em JAVA de um módulo usado para se realizar as consultas em bancos de dados externos a partir da propriedade especificada no arquivo de configuração.

## 5 EXPERIMENTOS E RESULTADOS

Neste capítulo, serão apresentados dois experimentos realizados a fim de se validar a abordagem MONET, bem como a sua implementação, o MONET Tool. Os resultados obtidos foram analisados sob a ótica da mineração de regras raras, foco do presente trabalho.

Cada um dos experimentos contou com um *dataset* local no formato N-Triples RDF, um *endpoint* sobre o qual foram executadas as consultas ao banco de dados externo da DBpedia, e os arquivos de configuração em XML que determinaram os parâmetros e métricas que foram utilizadas na execução do MONET Tool daquele experimento.

Assim, cada módulo de algoritmo de mineração presente no MONET Tool realizou, separadamente, uma mineração no *dataset* e teve como resultado um conjunto de regras  $R_{conjunto}^1 = \{R_1^1, R_2^1, \dots, R_n^1\}$ . Com isso, foi possível buscar, no *endpoint* externo da DBpedia, informações pré-determinadas no arquivo de configuração sobre os recursos que geraram essas regras  $S(R_{conjunto}^1) = \bigcup_{i=1}^n \mathcal{S}(R_i^1)$ .

Após esse passo, as informações externas foram inseridas no conjunto de dados inicial e, cada algoritmo, realizou o processo de mineração de regras novamente, considerando a modificação do valor do suporte mínimo mencionada em 4.1. Assim, foi obtido um novo conjunto de regras  $R_{conjunto}^2 = \{R_1^2, R_2^2, \dots, R_n^2\}$  que foi filtrado de maneira que só restaram aquelas que aparecem no segundo conjunto de regras, mas não no primeiro, ou seja,  $R_{conjunto}^3 = R_{conjunto}^2 / R_{conjunto}^1$ .

Vale ressaltar que os suportes das regras mostradas nas tabelas dos experimentos 5.1 e 5.2 estão considerando o aumento do número de nós do *dataset*, como explicado em 4.1, podendo os seus valores serem menores do que o suporte mínimo inicialmente definido. Além disso, pelo fato das regras mineradas pelo MRARE possuírem sempre confiança máxima, sendo sempre válidas, as informações sobre confiança foram ocultadas das tabelas de resultado desse algoritmo presentes nesses mesmos experimentos.

O foco do primeiro experimento foi a ferramenta MONET Tool e a validação da sua funcionalidade, apresentando uma comparação entre as regras que são obtidas utilizando-se tanto o MRAR quanto o MRARE implementados no MONET Tool. Um *dataset* sintético e reduzido foi utilizado por ser mais fácil analisar as saídas obtidas e verificar a veracidade das regras de associação encontradas e o *endpoint* genérico da DBpedia. O conjunto de dados escolhido foi o Dt\_Futebol, *dataset* baseado no que já foi utilizado no trabalho (5) e guarda informações relacionadas a jogadores de futebol. Ao final foram comparados os resultados obtidos utilizando tanto o algoritmo MRAR quanto o MRARE juntamente com o resto da abordagem MONET.

Por fim, o segundo experimento tem como objetivo mostrar o funcionamento do MRARE juntamente com a consulta a bancos externos em um grande conjunto de dados, mostrando como ele pode complementar o resultado do MRAR nesses tipos de *datasets*. O conjunto de dados utilizado foi uma adaptação do conjunto de dados aberto de botânica do Jardim Botânico do Rio de Janeiro, o Jabot.

## 5.1 Experimento com um *dataset* sintético

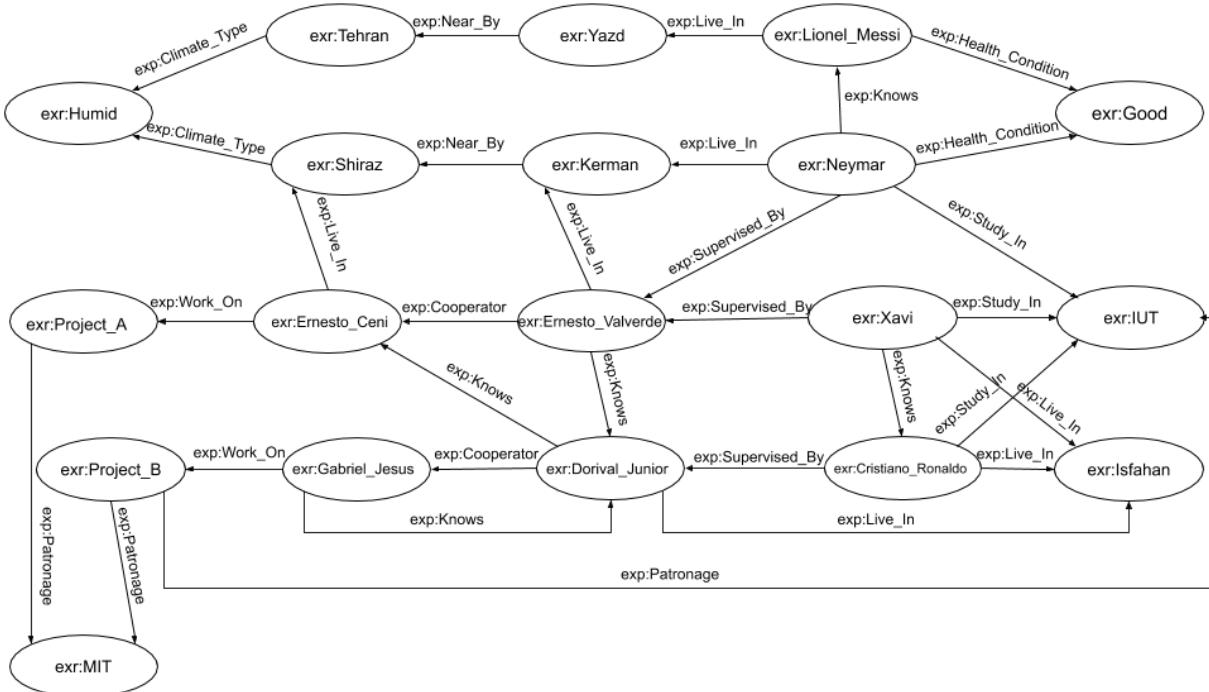
Neste primeiro experimento, focamos em uma comparação entre os resultados obtidos pelo MRAR e pelo MRARE incluindo a parte da abordagem que envolve consultas externas que foi implementada no MONET Tool atuando sobre um mesmo *dataset*. Mais que isso, as comparações feitas nesse experimento buscaram analisar as características únicas de cada abordagem de mineração de regras atuando sobre as mesmas *ItemChains*.

Para isso, foi necessário utilizar um suporte mínimo para o MRARE diferente do utilizado com o MRAR, uma vez que um *ItemChain* precisa ser raro para o MRARE e, ao mesmo tempo, frequente para o MRAR a fim de que ele seja explorado em ambos os algoritmos. Desse modo, o presente experimento não tem a intenção de encontrar o melhor algoritmo de mineração de regras, e sim explorar as suas semelhanças e diferenças na forma como as suas regras são apresentadas.

Para que fosse possível analisar os dois resultados obtidos, escolhemos um *dataset* reduzido e que foi baseado no *dataset* Dt\_Neymar utilizado por (5) no trabalho 3.1, chamado aqui de Dt\_Futebol. Esse conjunto de dados possui 24 nós e 36 arestas, que estão apresentado na Figura 12, a menos dos links *owl:sameAs* existentes entre o *endpoint* da DBpedia e os recursos: *exr:Lionel\_Messi*, *exr:Neymar*, *exr:Cristiano\_Ronaldo*, *exr:Xavi* e *exr:Gabriel\_Jesus*.

A consulta externa foi realizada no *endpoint* da DBpedia por ser um *dataset* grande e genérico, possuindo informações sobre os jogadores de futebol que estão descritos no Dt\_Futebol. As informações extraídas da base de dados externa foram as mesmas do trabalho (5), onde foram resgatados os times de futebol dos quais os jogadores presentes no *dataset* já participaram. Essa pesquisa foi feita através do predicado <http://dbpedia.org/ontology/team>, que se mostrou promissor na geração de regras com o *dataset* Dt\_Futebol.

Assim, como primeira etapa, o MRAR foi executado com as métricas de suporte mínimo igual a 8%, confiança mínima igual a 70%, e níveis mínimo e máximo de relações em cada *ItemChain* respectivamente iguais a 1 e 3. O arquivo de configuração utilizado está apresentado no apêndice A, onde mostramos todas as *tags* que foram utilizadas no experimento.



PREFIX exr: <<http://ex.com/resource/>>  
PREFIX exp: <<http://ex.com/predicate/>>

Figura 12 – Dataset Dt\_Futebol baseado no Dt\_Neymar de 3.1

Como resultado da primeira mineração, foram obtidas 13.538 regras de associação de multirrelação frequentes, com suporte igual a 8,3%, onde foram indicados cinco recursos que deram suporte a uma ou mais dessas regras para que eles fossem consultados do banco de dados externo, sendo eles: exr:Xavi, exr:Lionel\_Messi, exr:Cristiano\_Ronaldo, exr:Gabriel\_Jesus, exr:Neymar. Com isso, após feitas a consulta externa e a inserção do seu resultado no conjunto de dados local, o *dataset* original foi expandido de 24 nós para 50, aumentando em mais de 100%.

Na segunda mineração com o MRAR, foram encontradas 77.780 regras, representando um aumento de, aproximadamente, 574%, sendo um número bastante expressivo. Contudo, dessas regras, 64.242 foram geradas pela primeira vez, ou seja, que não estão contidas nas 13.538 regras gerada anteriormente, sendo 4 delas explicitadas na Tabela 1.

Analizando a Tabela 1, vemos que as regras 1 e 2 significam que "Se uma pessoa tem condição de saúde boa, ela mora perto de lugar com clima úmido e joga ou já jogou no Barcelona FC". Por outro lado, a regra 3 significa que "Se alguém mora perto de lugar com clima úmido e joga ou já jogou no Barcelona FC, esse alguém tem condição de saúde boa."

Por fim, a regra 4 significa que "Se alguém atende todos as cinco restrições em forma de *ItemChains* do antecedente, esse alguém é supervisionado por quem tenha um

Tabela 1 – Resultado da execução do MONET Tool com o MRAR no conjunto de dados do Dt\_Futebol

Regra	Antecedente	Consequente	Supp	Conf
1	exp:Health_Condition (exr:Good)	dbo:team (dbr:FC_Barcelona)	4%	100%
2	exp:Health_Condition (exr:Good)	exp:Live_In (exp:Near_By (exp:Climate_Type (exr:Humid)))	4%	100%
3	exp:Live_In (exp:Near_By (exp:Climate_Type (exr:Humid))), dbo:team (dbr:FC_Barcelona)	exp:Health_Condition (exr:Good)	4%	100%
4	exp:Supervised_By (exp:Knows (exp:Cooperator (exr:Gabriel_Jesus))), exp:Supervised_By (exp:Live_In (exr:Kerman)), exp:Supervised_By (exp:Knows (exr:Dorival_Junior)), exp:Study_In (exr:IUT), dbo:team (dbr:FC_Barcelona)	exp:Supervised_By (exp:Cooperator (exp:Work_On (exr:Project_A)))	4%	100%

cooperador que trabalha no projeto A". Pode-se perceber que essa última regra é muito extensa e pode apresentar dificuldades de ser aplicada em casos reais por possuir muitas restrições a serem atendidas para que a inferência do consequente da regra aconteça de fato.

Prosseguindo com o presente experimento, para que seja possível comparar as regras do MRAR e do MRARE com as mesmas *ItemChains*, o MRARE explorar os mesmos *ItemChains* que o MRAR explorou em suas primeiras regras mineradas com 8,3% de suporte, o que faz com que a métrica de suporte mínimo do MRARE seja maior que esse valor. Com isso, a implementação do MRARE foi rodada com a métrica de suporte mínimo igual a 9% e com o número mínimo e máximo de relações em cada *ItemChain* iguais a 1 e 3, respectivamente. O arquivo de configuração utilizado está apresentado no apêndice B, contendo as *tags* necessárias para a execução do presente experimento utilizando o MRARE.

Após a primeira mineração, foram geradas 21 regras de associação de multirrelação raras, valendo destacar que, pelo algoritmo MRARE, todas essas regras possuem confiança igual a 100%, configurando regras sempre verdadeiras. Desta vez, porém, foram identificados somente quatro recursos que deram suporte a uma ou mais dessas regras e que foram consultados no banco de dados externo, sendo eles: exr:Xavi, exr:Lionel\_Messi, exr:Gabriel\_Jesus, exr:Neymar. Assim, o *dataset*, que continha 24 nós, passou a ter 42, representando um aumento de 75% no número total de nós.

A segunda mineração feita, também com o MRARE, usou o conjunto de dados de 42 nós e gerou 28 regras, o que representa um aumento de mais de 33% nas regras geradas. Por fim, ao comparar essas regras com as 21 iniciais, descobriu-se que, das 28 regras, 26 são inéditas, ou seja, que não existiam antes da expansão do conjunto de dados local, sendo duas delas apresentadas na Tabela 2.

Inicialmente, podemos perceber que a regra 1 da Tabela 2 de resultados do MONET com o MRARE é igual a junção das regras 1 e 2 da Tabela 1 de resultados do MONET com o uso do MRAR, mostrando que o MRARE possibilitou uma inferência idêntica sobre a mesma informação utilizando um menor número de regras. Ao estudar mais a fundo

Tabela 2 – Resultado da execução do MONET Tool com o MRARE no conjunto de dados do Dt\_Futebol

Regra	Antecedente	Consequente	Supp
1	exp:Health_Condition (exr:Good)	dbo:team (dbr:FC_Barcelona), exp:Live_In (exp:Near_By (exp:Climate_Type (exr:Humid)))	4,76%
2	exp:Supervised_By (exp:Live_In (exr:Kerman))	exp:Supervised_By (exp:Cooperator (exp:Work_On (exr:Project_A))), exp:Supervised_By (exp:Knows (exp:Knows (exr:Rogério_Ceni))), exp:Supervised_By (exp:Cooperator (exr:Rogério_Ceni)), exp:Supervised_By (exp:Cooperator (exp:Live_In (exr:Shiraz))) exp:Supervised_By (exr:Ernesto_Valverde), exp:Supervised_By (exp:Live_In (exp:Near_IBy (exr:Shiraz))), exp:Study_In (exr:IUT), exp:Supervised_By (exp:Knows (exp:Live_In (exr:Isfahan))), exp:Supervised_By (exp:Knows (exr:Dorival_Junior)), exp:Supervised_By (exp:Knows (exp:Cooperator (exr:Gabriel_Jesus))), dbo:team (dbr:FC_Barcelona)	4,76%

essas regras, vemos que essa inferência de informações só acontece com aquelas regras do MRAR que contém somente uma *ItemChain* tanto no antecedente quanto no consequente, uma vez que o MRARE consegue gerar regras cujo formato consiste em uma *ItemChain* levando em várias *ItemChains* e -assim- pode unir um conjunto de regras do MRAR em uma só.

Analizando, agora, a regra 2 da Tabela 2 de resultados do MRARE, percebemos que a regra envolve uma *ItemChain* levando em onze *ItemChains*, o que significa que se um recurso atende ao *ItemChain* do antecedente, todos as dez *ItemChains* do consequente acontecem nesse recurso. Comparativamente com a regra 4 da Tabela 1 de resultados do MRAR, essa regra do MRARE é mais simples de ser aplicada, pois é mais fácil encontrar uma situação envolvendo uma única restrição que leva em dez consequências diretas do que encontrar uma situação envolvendo sete restrições que levam a somente uma consequência direta.

Por último, percebemos que existem regras que o MRAR gera e que o MRARE não é capaz de minerar, como, por exemplo, as regras 3 e 4 da Tabela 1, onde as regras possuem mais de uma *ItemChain* em seus antecedentes. Isso acontece porque o algoritmo do MRARE, ao trabalhar com *ItemChains* geradoras em classes de equivalência, procura criar regras mais genéricas, uma vez que a escolha por aquelas mais específicas, como as 3 e 4 mencionadas, podem ampliar muito o número total de regras, podendo causar a incapacidade de minerá-las por falta de recursos computacionais.

Após as análises feitas acima, podemos ver que as regras obtidas com o uso da ferramenta MONET Tools associada a ferramenta MRARE se mostraram mais simples de serem aplicadas que as obtidas com o uso do MRAR, pois, a partir de somente uma *ItemChain* no antecedente, as regras do MRARE conseguem apresentar várias inferências.

Por outro lado, também pudemos observar que o uso do MONET Tools com o MRARE não consegue gerar todas as regras que a sua utilização com o MRAR é capaz, uma vez que o MRAR pode inserir mais de uma *ItemChain* no antecedente para inferir

alguma informação, o que o leva a criar algumas regras muito mais específicas.

A redução do número de regras do MRARE aliada ao fato de que ele é capaz de unir diversas regras do MRAR numa única regra, faz com que o MONET associado ao MRARE gere um menor número de regras, como é mostrado no presente experimento, onde o MONET com o MRAR minerou 64.242 regras novas, enquanto que o MONET com o MRARE minerou somente 26 regras inéditas, representando uma redução de aproximadamente 96% no número de regras. Isso é um reflexo direto da utilização do conceito de *ItemChains* geradoras no algoritmo do MRARE, fazendo que com as suas regras sejam muito mais agregadas, contendo mais inferências por regras mineradas.

Por fim, foi verificado que o algoritmo do MRARE, rodando com suporte mínimo de 9%, conseguiu alcançar regras com suporte de 8,3%, o que demonstra a sua eficácia no objetivo de minerar regras raras, segundo a definição 2.2.6. Como o *dataset* utilizado no presente experimento é pequeno e, portanto, possui um número limitado de nós, quando calculamos o suporte das regras geradas para esse *dataset*, observamos que ele não possui dados suficientes para que possam existir regras com suporte semelhante àqueles encontrados na mineração em bancos reais, onde muitas das regras possuem suporte com ordem de grandeza inferior a  $10^{-3}$ . Assim, devido aos seus suportes serem grandes o suficiente, o MRAR conseguiu alcançar todas as regras mineradas pelo MRARE.

## 5.2 Experimento com o JabotG

Esse experimento realiza as mesmas etapas do experimento 5.1 e tem como objetivo mostrar a eficácia da utilização do MONET Tool em um banco de dados real, onde o ambiente é muito menos controlado que o *dataset* sintético Dt\_futebol utilizado no experimento 5.1. Para isso, foram utilizados dados legítimos provenientes do banco de dados relacional de coleções científicas do Jardim Botânico do Rio de Janeiro, o Jabot, que possui mais de 790 mil tuplas segundo o que diz (5).

De acordo com (12), esse banco de dados reflete o conhecimento adquirido de botânicos sobre coleções botânicas, onde pesquisadores realizaram diversas expedições por todo o Brasil para coletar dados, além do intercâmbio com outros institutos nacionais e internacionais, demonstrando ser um acervo de grande importância científica para o conhecimento da flora do Brasil.

Para ser possível a aplicação do MONET Tool nesse conjunto de dados, foi necessário utilizar, na verdade, uma adaptação do Jabot para o formato de um grafo unidirecional denominada JabotG. Existe uma tendência mundial, que já foi mencionada anteriormente neste trabalho, de se padronizar o formato dos dados disponíveis na Web para facilitar a interoperabilidade deles.

Nesse sentido, Oliveira et al.(5) realizou em seu trabalho (5) essa transformação do banco de dados do Jardim Botânico do formato relacional para o de grafo direcionado que possibilitou que informações externas sobre os dados que já constavam no banco fossem buscadas e anexadas no *dataset*, enriquecendo-o e possibilitando o alcance de novas regras importantes. A modelagem em grafo do JabotG está apresentada na Figura 13 para melhor compreensão do leitor.

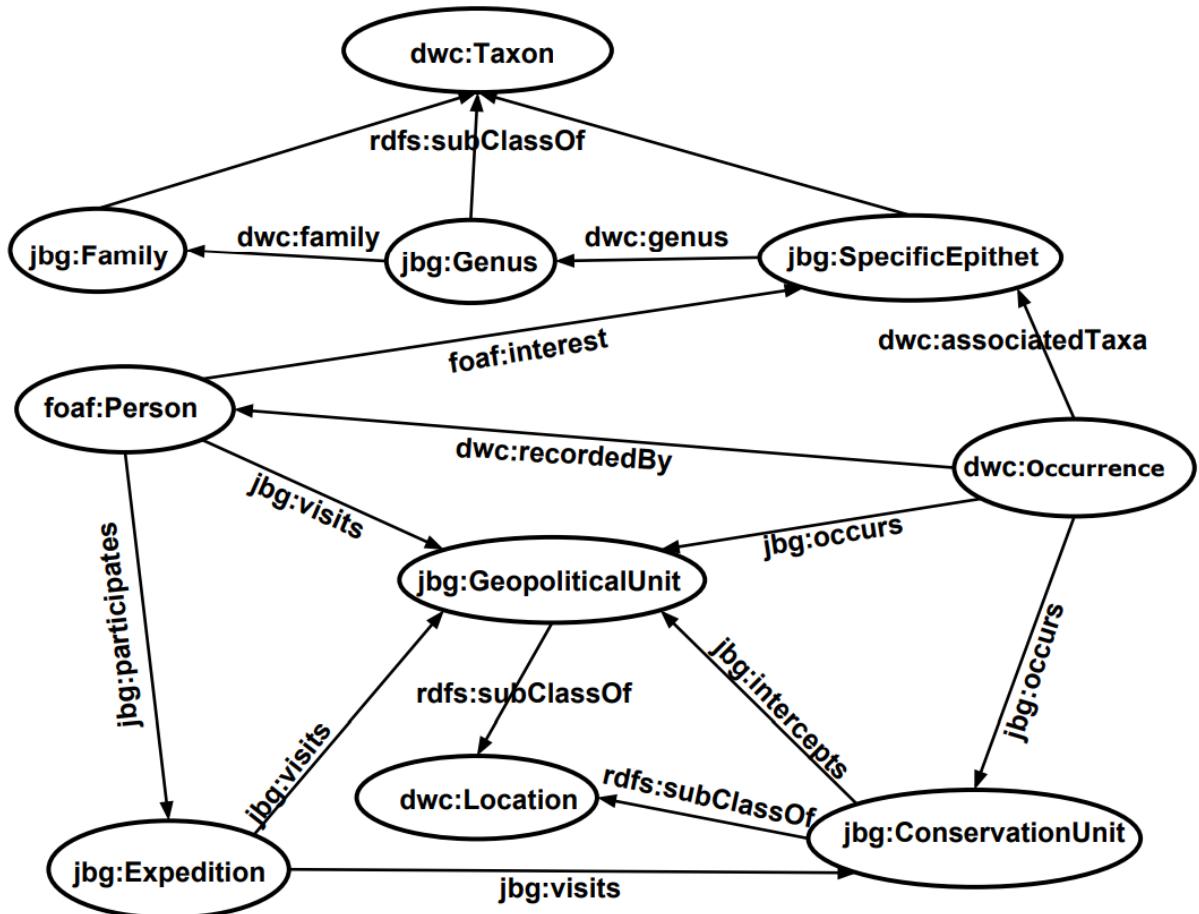


Figura 13 – Modelo em grafo do JabotG. Fonte: (5)

Para testar a ferramenta MONET Tools, foi definido que seria um objetivo para a sua execução relacionar as espécies presentes no conjunto de dados JabotG com dados externos relacionados ao clima das cidades onde existem ocorrências dessas espécies. Como fonte dos dados externos, *endpoint*, desse experimento foi escolhida a versão brasileira da DBpedia, a DBpedia-pt, por possuir informações relevantes sobre o tipo de clima dos recursos de cidade brasileiras descritos no JabotG.

Algumas adaptações tiveram que ser feitas no modelo proposto do JabotG para possibilitar a execução desse experimento. Isso ocorreu porque as regras são diretamente influenciadas pelo *schema* do grafo utilizado como entrada do algoritmo, sendo assim, como nesse experimento desejamos relacionar os recursos de espécie presentes no JabotG

com os climas das cidades onde elas ocorrem e que estão descritas no banco de dados externo da DBpedia-pt, foi preciso que o *schema* fosse modificado para que as cidades fossem os recursos que dão suporte às regras mineradas.

O modelo utilizado está apresentado na Figura 14, onde buscamos criar caminhos no grafo que formassem *ItemChains* entre cidades e espécies. Nessa Figura, é possível verificar que a cidade está em um extremo do grafo, a partir do qual vários caminhos são formados. Assim, existe uma maior possibilidade dessas cidades fornecerem suporte às regras mineradas pelos algoritmos.

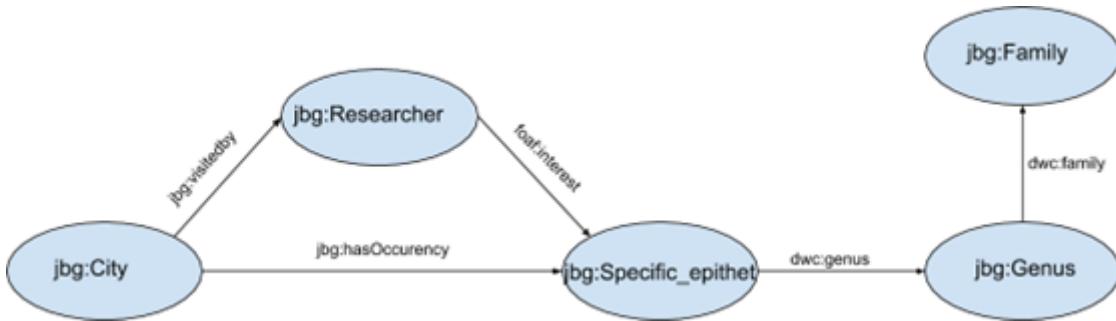


Figura 14 – Modelo em grafo do JabolG adaptado

Além disso, para realizar a consulta externa, foram descobertos *links* entre 1.714 recursos de cidades presentes nesse conjunto de dados com a DBpedia-pt, o que foi feito através da ferramenta LIMES, e esses *links* foram inseridos no conjunto de dados inicial, que possui 111.591 nós, totalizando 114.955 nós. Por fim, foi realizado um recorte no *dataset* para que ele contivesse somente as cidades com ligações externas, passando a ter somente 25.118 nós, o que representa uma redução de, aproximadamente, 78% no número total de nós do grafo.

Como primeira etapa desse experimento, foi executado o MONET em conjunto com o algoritmo MRAR no *dataset* com 25.118 nós, com o suporte mínimo de 3%, confiança mínima de 70%, e níveis mínimo e máximo de *ItemChains*, respectivamente, iguais a 1 e 3, além das informações sobre a consulta externa, como mostrado no apêndice C. É importante ressaltar que, tratando-se de regras frequentes, 3% de suporte mínimo já é um valor bastante baixo. Valores ainda menores do que esses causam muita demora na execução do MONET, pois o número de *ItemChains* com o suporte maior ou igual a esse é muito elevado. Então, devido às limitações computacionais, esse valor se revelou um limite de execução do MONET Tool com a utilização do MRAR.

Duramente a primeira mineração de regras, foram encontradas 7.570 regras e 1.602 recursos cidade como suporte dessas regras e que foram usados para realizar as consultas externas na DBpedia-pt. Após a consulta externa e a união do seu resultado ao conjunto de dados inicial, o número de nós passou de 25.118 para 25.385, representando um aumento de aproximadamente 1%.

A segunda mineração de regras com o MRAR foi realizada, tendo sido encontradas também 7.570 regras e, quando comparadas com as da primeira mineração, foi descoberto que se tratavam das mesmas regras, ou seja, o processo de consulta externa e união dos resultados no *dataset* inicial não resultou em novas regras frequentes com o uso do algoritmo MRAR.

Continuando o experimento, na segunda etapa, foi executado o MONET Tool em conjunto com o algoritmo MRARE no *dataset* adaptado JabotG com 25.118 nós. Como mencionamos anteriormente, a execução do MONET Tool utilizando o MRAR com suporte mínimo menor que 3% se torna inviável computacionalmente devido à grande quantidade de regras que serão geradas, na execução do MONET Tool com o MRARE, encontramos uma questão parecida quando utilizamos suportes mínimos maiores que 0,01%. Quando essa configuração é utilizada, é necessário muito tempo e espaço em memória para que a execução possa ser finalizada.

Assim, as entradas utilizadas para a etapa de execução do MONET Tool com o MRARE foram iguais a 3% para o suporte mínimo, 1 para o nível mínimo e 3 para o nível máximo de cada *ItemChains*. Além disso, a descrição da consulta usada no banco de dados externo, hospedado localmente também está presente no arquivo de configuração, apresentado no Apêndice D.

Durante a primeira mineração de regras, foram encontradas 8.031 regras e 1.000 recursos cidade como suporte dessas regras e que foram usados para as consultas externas na DBpedia-pt. Após a consulta externa e a agregação do seu resultado ao conjunto de dados, o número de nós passou de 25.118 para 25.297, representando um aumento de 0,7% aproximadamente.

A segunda mineração de regras foi realizada, agora com o MRARE, tendo sido encontradas 8.045 regras que representam um aumento de 0,17% no número de regras aproximadamente. Quando comparadas com as da primeira mineração, foi descoberto que 611 delas se tratava de regras diferentes, ou seja, 611 regras não existiam antes do processo de consulta externa e agregação dos resultados ao *dataset*.

Após pesquisas e conversas com especialistas biólogos, pudemos separar a análise dos nossos resultados em duas categorias que serão discutidas separadamente a seguir.

### 5.2.1 Espécies Especialistas

Nesta categoria iremos mostrar regras que envolvem espécies que são ditas especialistas, pois vivem em um habitat restrito e necessitam de determinados recursos para sobreviver possuindo um nicho ecológico bastante restrito, segundo o que diz SANTOS(25). Como exemplo de regras encontradas envolvendo essas espécies, temos duas delas apresentadas na Tabela 3.

Tabela 3 – Resultado da execução do MONET Tool com o MRARE no conjunto de dados do JabotG

Regra	Antecedente	Consequente	Supp
1	jbg:hasOccurrency (jbg:vanhouttea_lanata)	dbp:clima (dbr:Clima_tropical_de_altitude)	0,007906%
2	jbg:hasOccurrency (jbg:orthophytum_diamantinense)	dbp:clima (dbr:Clima_tropical_de_altitude)	0,007906%

As espécies descritas na Tabela 3, *Vanhouttea lanata* e *Orthophytum diamantinense*, segundo (26), são espécies endêmicas, o que significa que elas ocorrem exclusivamente em uma determinada região geográfica, configurando uma espécie especialista.

Para a primeira regra, o clima associado às ocorrências da espécie é o clima tropical de altitude. No anexo A, temos um mapa seguindo a classificação climática de Köppen-Geiger que é bastante utilizada em geografia, climatologia e ecologia. Por (27), podemos fazer uma associação direta entre os climas do tipo Cw presentes no anexo A e o clima tropical de altitude que foi encontrado pelo processo de mineração. As ocorrências reais da espécie *Vanhouttea Lanata* estão na região serrana do Rio de Janeiro e estão indicadas na Figura 15 sendo possível cruzar essa informação com o mapa de climas e validar a regra.



Figura 15 – Localizações de ocorrências da espécie de planta *Vanhouttea lanata*, extraída de (6).

Considerando agora a segunda regra mostrada, mais uma vez o algoritmo se mostrou eficiente em associar a espécie especialista *Orthophytum diamantinense* ao clima tropical de altitude, como podemos confirmar cruzando as ocorrências reais dessa espécie encontradas na Figura 16 com as informações de clima do Brasil do anexo A .

Essas duas regras apresentadas acima dão uma indicação de que a nossa ferramenta consegue resultados satisfatórios para espécies especialistas. Não é possível verificar todas



Figura 16 – Localizações de ocorrências da espécie de planta *Orthophytum diamantinense*, extraída de (6).

as ocorrências desse tipo de espécie que estão presentes nas regras geradas devido ao fato de termos utilizados um banco real com um número de entradas muito grandes, mas conseguimos ter um bom indicativo da utilidade da ferramenta observando apenas algumas dessas entradas.

### 5.2.2 Espécies Generalistas e Gêneros

Vamos agora discutir o segundo conjunto de regras, que envolve as espécies ditas generalistas, o que significa dizer que elas são pouco seletivas, que podem viver em quase qualquer ambiente e estão dispersas pelo Brasil, possuindo um Nicho Ecológico bastante amplo (25). Além disso, nesse conjunto vamos também abordar regras que envolvem o gênero e não espécies de planta, informação que também está disponível no bando de dados JabotG. A seguir, vamos apresentar alguns exemplos desses tipos de regras que foram encontrados através da execução do MONET Tool com foco em regras raras.

Tabela 4 – Resultado da execução do MONET Tool com o MRARE no conjunto de dados do JabotG

Regra	Antecedente	Consequente	Supp
1	jbg:hasOccurrency (jbg:urochloa_decumbens)	dbp:clima (dbr:Clima_subtropical)	0,007906%
2	jbg:hasOccurrency (jbg:cassia_fistula))	dbp:clima (dbr:Clima_tropical)	0,007906%
3	jbg:hasOccurrency (dwc:genus (jbg:verbenaceae_privaria))	dbp:clima (dbr:Clima_tropical)	0,007906%
4	jbg:hasOccurrency (dwc:genus (jbg:phyllanthaceae_amanoa))	dbp:clima (dbr:Clima_tropical)	0,007906%

Pela Figura 17, pode-se ver que as espécies referentes às regras 1 e 2 da Tabela 4, respectivamente *Urochloa decumbens* e *Cassia fistula*, ocorrem em muitas regiões do Brasil. Isso significa que dificilmente poderíamos chegar a uma regra associando tais espécies a um clima específico. Contudo, regras como essas acabaram sendo geradas pelo algoritmo, demonstrando uma imprecisão envolvendo espécies não endêmicas.

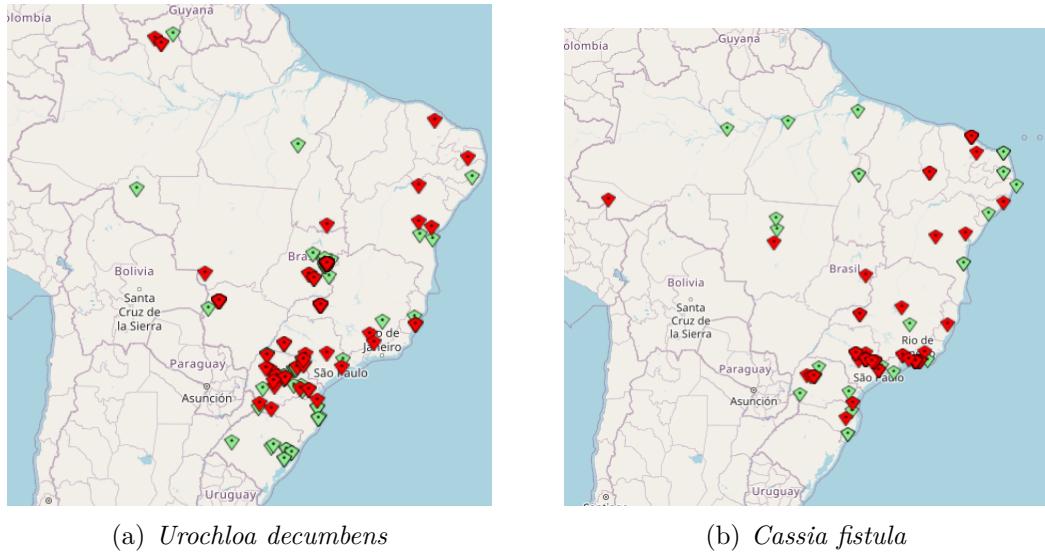


Figura 17 – Localizações de ocorrências das espécies de plantas, extraídas de (6).

Esse erro pode estar associado à redução do *dataset* realizada para que apenas as cidades com ligações de similaridade do tipo *owl:sameAs* com a DBpedia-pt estivessem incluídas no conjunto de dados. A ideia foi reduzir o dataset para viabilizar os experimentos, já que o processamento é custoso. Assim, ficaram de fora do dataset algumas cidades, e entre elas as cidades com climas diferentes onde também ocorrem as espécies não endêmicas, que aparecem nas regras mineradas (regras 1 e 2). Esse erro encontrado, poderia ser contornado se for possível realizar um maior número de ligações entre os recursos do tipo cidade do JabotG e a DBpedia-pt, constituindo-se um conjunto de dados mais completo.

Da mesma forma, as regras 3 e 4 da Tabela 4 mostram inferências de ocorrências de gêneros em cidades com clima tropical. Mesmo antes de verificar o mapa das localizações de ocorrências apresentado na Figura 18 e compará-lo com o mapa de climas do Brasil presente no anexo A, podemos discutir a veracidade de tal regra. Isso porque um gênero pode estar associado a diversas espécies, e, portanto, a várias localidades distintas, o que nos leva a pensar que as regras encontradas são imprecisas.

Ao confrontar as ocorrências dos gêneros com o mapa de climas do Brasil, confirmamos a suposição de que essas regras são imprecisas. Antes de executar o algoritmo no *dataset* JabotG, realizamos um recorte nesse banco de dados para que só restassem cidades cujas informações externas constavam na DBpedia-pt, nesse processo foram descartadas diversas cidades que gerariam regras caso não houvesse a etapa da busca externa. Assim,

na inferência relacionada aos climas onde cada espécie ocorre, algumas cidades onde essa relação de ocorrência seria verdadeira não participaram do processo de mineração, o que resultou nas regras incompletas.

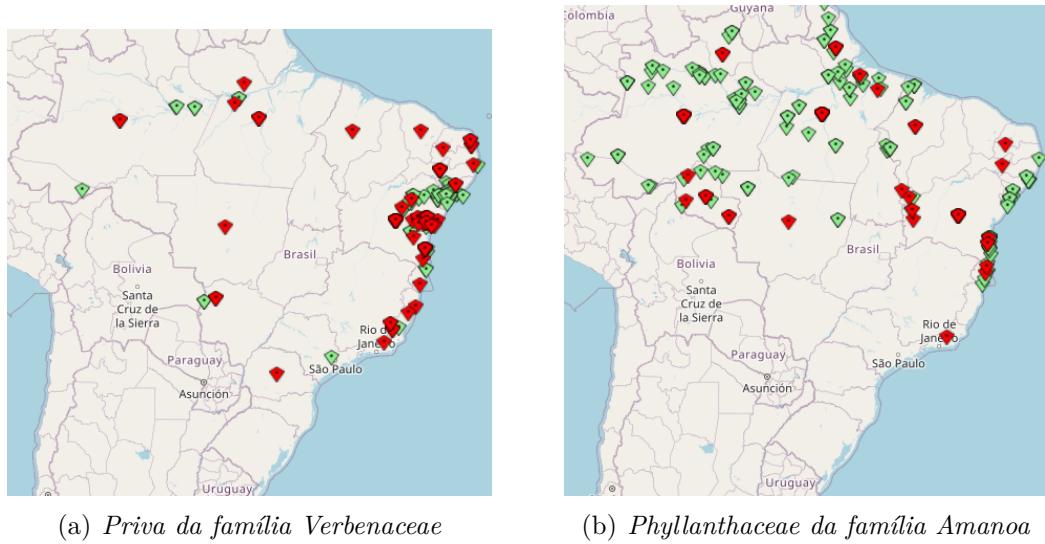


Figura 18 – Localizações de ocorrências dos gêneros de plantas, extraídas de (6).

### 5.2.3 Conclusão sobre o experimento com o JabotG

Observando os dois grupos de experimentos, podemos concluir que o MONET Tool utilizado juntamente com a implementação do MRARE oferece resultados satisfatórios. As regras encontradas associadas às espécies especialistas se mostraram condizentes com a realidade no sentido de que os climas encontrados para as ocorrências dessa espécie são de fato os climas onde elas ocorrem no ambiente segundo o *site* da Flora do Brasil<sup>1</sup>.

Já as regras relacionadas às espécies generalistas e aos gêneros estão, de fato, incompletas. Porém, não foram detectadas inconsistências no sentido de estarmos associando essas ocorrências a climas equivocados. Apenas não foram encontrados todos os climas associados a elas. Dessa forma, tudo nos leva a crer que a inconsistência foi causada pelo recorte do *dataset* que foi adotado para possibilitar que a ferramenta fosse executada em tempo viável.

Além disso, notou-se que as regras com suporte entre 3% e 0,01% não foram exploradas por nenhum dos dois algoritmos, já que o MRAR lidou com aquelas cujo suporte era superior ao primeiro valor e o MRARE lidou com aquelas cujo suporte era inferior ao segundo valor. Assim, podemos dizer que o MONET Tool não exauriu os possíveis resultados desse experimento, deixando em aberto a possibilidade da existência de regras cujos suportes encontram-se entre esses dois valores.

<sup>1</sup> <http://floradobrasil.jbrj.gov.br/>

Por fim, esse experimento demonstrou que o MRARE consegue complementar os resultados do MRAR, uma vez que ele encontrou regras com suportes baixos que o MRAR não foi capaz de minerar. Isso porque quanto mais baixo o suporte, mais *ItemChains* são geradas e analisadas pelo algoritmo do MRAR, o que pode resultar em um alto custo computacional em termos de tempo e memória para a sua execução.

## 6 CONCLUSÃO

Esse trabalho foi idealizado com o objetivo de criar uma ferramenta capaz de enriquecer um conjunto de dados alvo e minerar regras de associação multirrelação que também levassem em consideração a raridade de entidades dentro desse *dataset*. Essa ferramenta foi baseada no MRAR+, envolvendo poucas mudanças nesse algoritmo e o acréscimo de outras funcionalidades. Dentro desse contexto, consideramos o objetivo desse trabalho alcançado.

O MRAR+ foi desenvolvido no Instituto Militar de Engenharia em uma tese de mestrado, contudo foi necessário estendê-lo para incluir a nova funcionalidade para a mineração envolvendo entidades raras. Várias soluções foram buscadas na literatura e aquela que melhor se adequava ao objetivo de mineração de regras raras foi escolhida por ser mais robusta e precisar de menor manipulação dos *datasets* para ser utilizada pelo usuário. Após isso, foi estudada a fundo a solução escolhida, sendo feito alterações para que ela fosse compatível em *datasets* baseados em grafos direcionados.

As contribuições desse trabalho foram:

- a especificação do formalismo da abordagem de mineração de regras de associação de multirrelação raras MRARE aplicada a grafos direcionais, fugindo da abordagem tradicional de bancos relacionais;
- a especificação da nova abordagem chamada MONET, além de sua implementação, o MONET Tool, feita na linguagem de programação JAVA, contendo o MRAR, MRAR+ e o MRARE;
- o estudo de caso com dois experimentos: o primeiro realizado com um banco de dados sintético onde foi possível analisar com mais facilidade a validade das regras geradas após a execução da nova ferramenta; o segundo teve como foco o *dataset* JabotG a fim de apresentar o comportamento satisfatório tanto do MRARE quanto do MONET Tool frente a um banco de dados real e sua quantidade enorme de dados, mostrando também que o MRARE consegue complementar o MRAR ao minerar regras com suportes muito baixos.

Por fim, sugerimos como contribuições futuras:

- a criação de uma interface gráfica onde seja possível ajustar os parâmetros de execução da ferramenta sem a necessidade de modificar manualmente o arquivo XML, e que também exiba as regras geradas de maneira visual e de fácil compreensão,

pois atualmente é preciso ter certa familiaridade com as saídas do programa para se entender o significado das regras geradas;

- a especificação e implementação de um algoritmo de mineração de regras de associação de multirrelação em *datasets* no formato RDF que minere as regras cujos suportes se encontram entre dois valores estipulados, sendo possível a análise de regras com suportes entre 0,01% e 3% do experimento 5.2;
- a especificação e implementação de um algoritmo de mineração de regras de associação de multirrelação em *datasets* no formato RDF que, diferentemente do trabalho 3.2, possuem suportes mínimos distintos para cada tipo de entidade do conjunto de dados, ou seja, cujos valores dependem das classes que as suas entidades possuem relações no *schema*
- a melhora da complexidade dos algoritmos de mineração de regras de associação de multirrelação MRAR e MRARE, uma vez que, no experimento 5.2, esses algoritmos foram computacionalmente custosos com o suporte mínimo menor que 3% e maior que 0,01%, respectivamente.

## REFERÊNCIAS

- 1 LOD. Linked Open Data (LOD). 09 mai. de 2020. Disponível em: <<https://lod-cloud.net/>>.
- 2 RAMEZANI, R.; SARAEE, M.; NEMATBAKHS, M. Mrar: Mining multi-relation association rules. Journal of Computing and Security, v. 1, p. 133–158, 01 2014.
- 3 OLIVEIRA, F. de. MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO DE MULTIRRELAÇÃO EM DATASETS NA WEB DE DADOS. Rio de Janeiro: [s.n.], 2018. 95 p. (Curso de Mestrado em Sistemas e Computação). 02 mar. de 2020. Disponível em: <<http://www.comp.ime.eb.br/pos/arquivos/publicacoes/dissertacoes/2018/2018-Felipe.pdf>>.
- 4 SZATHMARY, L.; VALTCHEV, P.; NAPOLI, A. Finding minimal rare itemsets and rare association rules. In: International Conference on Knowledge Science, Engineering and Management. [S.l.: s.n.], 2010. v. 6291, p. 16–27.
- 5 OLIVEIRA, F. A. de; COSTA, R. L.; GOLDSCHMIDT, R. R.; CAVALCANTI, M. C. Multirelation association rule mining on datasets of the web of data. In: Proceedings of the XV Brazilian Symposium on Information Systems, SBSI 2019, Aracaju, Brazil, May 20-24, 2019. [S.l.]: ACM, 2019. p. 61:1–61:8.
- 6 JBRJ. Flora do Brasil. 15 out. de 2020. Disponível em: <<http://floradobrasil.jbrj.gov.br/>>.
- 7 ALVARES, C. A.; STAPE, J. L.; SENTELHAS, P. C.; GONÇALVES, J. L. de M.; SPAROVEK, G. Köppen's climate classification map for brazil. Meteorologische Zeitschrift, E. Schweizerbart'sche Verlagsbuchhandlung, v. 22, n. 6, p. 711–728, 2013.
- 8 NGOMO, A.-C. N.; AUER, S. Limes – a time-efficient approach for large-scale link discovery on the web of data. Proceedings of IJCAI, "", p. 2312–2317, 01 2011.
- 9 LIMES, D. G. LIMES - Link Discovery Framework for Metric Spaces. 10 mai. de 2020. Disponível em: <<https://github.com/dice-group/limes>>.
- 10 DEER, D. G. The RDF Dataset Enrichment Framework. 10 mai. de 2020. Disponível em: <<https://github.com/dice-group/deer>>.
- 11 FOX, D. G. FOX - Federated Knowledge Extraction Framework. 10 mai. de 2020. Disponível em: <<https://github.com/dice-group/fox>>.
- 12 JANEIRO, J. B. do Rio de. Coleções Biológicas do Jardim Botânico do Rio de Janeiro. 10 out. de 2020. Disponível em: <<http://www.jbrj.gov.br/colecoes/biologicas>>.
- 13 SEGUNDO, J. E. S.; CONEGLIAN, C. S.; LUCAS, E. R. d. O. Conceitos e tecnologias da web semântica no contexto da colaboração acadêmico-científica: um estudo da plataforma vivo. Transinformação, SciELO Brasil, v. 29, n. 3, p. 297–309, 2017.
- 14 DIAS, T. D.; SANTOS, N. Web semântica: conceitos básicos e tecnologias associadas. Cadernos do IME-Série Informática, v. 14, p. 80–92, 2003.

- 15 FERREIRA, J. A.; SANTOS, P. L. V. A. d. C. O modelo de dados resource description framework (rdf) e o seu papel na descrição de recursos. *Informação* amp; Sociedade: Estudos, v. 23, n. 2, jul. 2013. 15 mai. 2020. Disponível em: <<https://periodicos.ufpb.br/ojs/index.php/ies/article/view/15436>>.
- 16 W3C. Semantic Web. 05 mai. de 2020. Disponível em: <<https://www.w3.org/standards/semanticweb>>.
- 17 FOAF. Friend of a Friend (FOAF) Ontology. 08 mai. de 2020. Disponível em: <<http://www.foaf-project.org/>>.
- 18 FUSEKI, A. J. Apache Jena Fuseki. 10 mai. de 2020. Disponível em: <<https://jena.apache.org/documentation/fuseki2/>>.
- 19 DBPEDIA. DBpedia. 10 mai. de 2020. Disponível em: <<https://wiki.dbpedia.org/>>.
- 20 GEONAMES. GeoNames Ontology. 10 mai. de 2020. Disponível em: <<https://www.geonames.org/>>.
- 21 DBPEDIA-PT. DBpedia-pt. 10 mai. de 2020. Disponível em: <<http://pt.dbpedia.org/>>.
- 22 LIU, B.; HSU, W.; MA, Y. Mining association rules with multiple minimum supports. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, 1999. p. 337–341. 02 mar. 2020. Disponível em: <<https://doi.org/10.1145/312129.312274>>.
- 23 VELOSO, A.; COUTINHO, B.; PÔSSAS, B.; MENEZES, G.; JR, W. M.; CARVALHO, M.; AMORIM, C. Mineração assíncrona de regras de associação em sistemas de memória compartilhada-distribuída. In: Anais do II Workshop em Sistemas Computacionais de Alto Desempenho. [S.l.: s.n.], 2001. p. 9–16.
- 24 SPECK, R.; NGOMO, A.-C. N. Named entity recognition using fox. In: Proceedings of the ISWC 2014, Posters & Demonstrations Track, CEUR Workshop Proceedings. [S.l.: s.n.], 2014. v. 1272, p. 85–88.
- 25 SANTOS, M. M. W. d. M. Perfil reprodutivo de plantas endêmicas de Caatinga, raras e ameaçadas de extinção ocorrentes no Parque Nacional do Catimbau, Pernambuco. 2016.
- 26 FLORA, C. Referência nacional em geração, coordenação e difusão de informação sobre biodiversidade e conservação da flora brasileira ameaçada de extinção. 15 out. de 2020. Disponível em: <<http://cncflora.jbrj.gov.br/>>.
- 27 MEDEIROS, R. M. de; CAVALCANTI, E. P.; DUARTE, J. F. de M. Classificação climática de köppen para o estado do piauí-brasil. REVISTA EQUADOR, v. 9, n. 3, p. 82–99, 2020.

# APÊNDICE A – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRAR EM DT\_FUTEBOL

Listing A.1 – Estrutura do arquivo de configuração

```

<?xml version="1.0" encoding="UTF-8"?>
<CONFIGURATION>
    <MODULE>
        <MRAR> ON </MRAR>
        <MRARE> OFF </MRARE>
        <LOD> ON </LOD>
    </MODULE>

    <RULE>
        <SUPPORT> 0.08 </SUPPORT>
        <CONFIDENCE> 0.7 </CONFIDENCE>
        <LEVEL>
            <MIN> 1 </MIN>
            <MAX> 3 </MAX>
        </LEVEL>
    </RULE>

    <INPUT>
        <SOURCE>
            <FILE> C:/PFC/data/Dt_Futebol.nt </FILE>
        </SOURCE>
        <TARGET>
            <PREVIEW> ON </PREVIEW>
            <ENDPOINT> http://dbpedia.org/sparql </ENDPOINT>
            <PREFIX>
                <LABEL> dbo </LABEL>
                <NAMESPACE> http://dbpedia.org/ontology/ </NAMESPACE>
            </PREFIX>
            <PROPERTY> dbo:team </PROPERTY>
        </TARGET>
    </INPUT>

    <OUTPUT>
        <MRAR>

```

```
<NAME> futebol_mrar_rules </NAME>
<PATH> C:/PFC/rules </PATH>
</MRAR>
</OUTPUT>
</CONFIGURATION>
```

## APÊNDICE B – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRARE EM DT\_FUTEBOL

Listing B.1 – Estrutura do arquivo de configuração

```

<?xml version="1.0" encoding="UTF-8"?>
<CONFIGURATION>
    <MODULE>
        <MRAR> OFF </MRAR>
        <MRARE> ON </MRARE>
        <LOD> ON </LOD>
    </MODULE>

    <RULE>
        <SUPPORT> 0.09 </SUPPORT>
        <LEVEL>
            <MIN> 1 </MIN>
            <MAX> 3 </MAX>
        </LEVEL>
    </RULE>

    <INPUT>
        <SOURCE>
            <FILE> C:/PFC/data/Dt_Futebol.nt </FILE>
        </SOURCE>
        <TARGET>
            <PREVIEW> ON </PREVIEW>
            <ENDPOINT> http://dbpedia.org/sparql </ENDPOINT>
            <PREFIX>
                <LABEL> dbo </LABEL>
                <NAMESPACE> http://dbpedia.org/ontology/ </NAMESPACE>
            </PREFIX>
            <PROPERTY> dbo:team </PROPERTY>
        </TARGET>
    </INPUT>

    <OUTPUT>
        <MRARE>
            <NAME> futebol_mrare_rules </NAME>

```

```
<PATH> C:/PFC/rules </PATH>
</MRARE>
</OUTPUT>
</CONFIGURATION>
```

## APÊNDICE C – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRAR EM DT\_JABOTG

Listing C.1 – Estrutura do arquivo de configuração

```

<?xml version="1.0" encoding="UTF-8"?>
<CONFIGURATION>
    <MODULE>
        <MRAR> ON </MRAR>
        <MRARE> OFF </MRARE>
        <LOD> ON </LOD>
    </MODULE>

    <RULE>
        <SUPPORT> 0.03 </SUPPORT>
        <CONFIDENCE> 0.7 </CONFIDENCE>
        <LEVEL>
            <MIN> 1 </MIN>
            <MAX> 3 </MAX>
        </LEVEL>
    </RULE>

    <INPUT>
        <SOURCE>
            <FILE> C:/PFC/data/Dt_JabotG_reduced.nt </FILE>
        </SOURCE>
        <TARGET>
            <PREVIEW> ON </PREVIEW>
            <ENDPOINT> http://localhost:3030/dbpediapt/sparql </ENDPOINT>
            <PREFIX>
                <LABEL> dbp </LABEL>
                <NAMESPACE> http://pt.dbpedia.org/property/ </NAMESPACE>
            </PREFIX>
            <PROPERTY> dbp:clima </PROPERTY>
        </TARGET>
    </INPUT>

    <OUTPUT>
        <MRAR>
            <NAME> rare_rules_mrare_jabot </NAME>

```

```
<PATH> C:/PFC/rules </PATH>
</MRAR>
</OUTPUT>
</CONFIGURATION>
```

## APÊNDICE D – ARQUIVO DE CONFIGURAÇÃO MONET TOOL COM MRARE EM DT\_JABOTG

Listing D.1 – Estrutura do arquivo de configuração

```

<?xml version="1.0" encoding="UTF-8"?>
<CONFIGURATION>
    <MODULE>
        <MRAR> OFF </MRAR>
        <MRARE> ON </MRARE>
        <LOD> ON </LOD>
    </MODULE>

    <RULE>
        <SUPPORT> 0.0001 </SUPPORT>
        <CONFIDENCE> 0.7 </CONFIDENCE>
        <LEVEL>
            <MIN> 1 </MIN>
            <MAX> 3 </MAX>
        </LEVEL>
    </RULE>

    <INPUT>
        <SOURCE>
            <FILE> C:/PFC/data/Dt_JabotG_reduced.nt </FILE>
        </SOURCE>
        <TARGET>
            <PREVIEW> ON </PREVIEW>
            <ENDPOINT> http://localhost:3030/dbpediapt/sparql </ENDPOINT>
            <PREFIX>
                <LABEL> dbp </LABEL>
                <NAMESPACE> http://pt.dbpedia.org/property/ </NAMESPACE>
            </PREFIX>
            <PROPERTY> dbp:clima </PROPERTY>
        </TARGET>
    </INPUT>

    <OUTPUT>
        <MRAR>
            <NAME> rare_rules_mrare_jabot </NAME>

```

```
<PATH> C:/PFC/rules </PATH>
</MRAR>
</OUTPUT>
</CONFIGURATION>
```

## ANEXO A – MAPA DE CLIMAS DO BRASIL

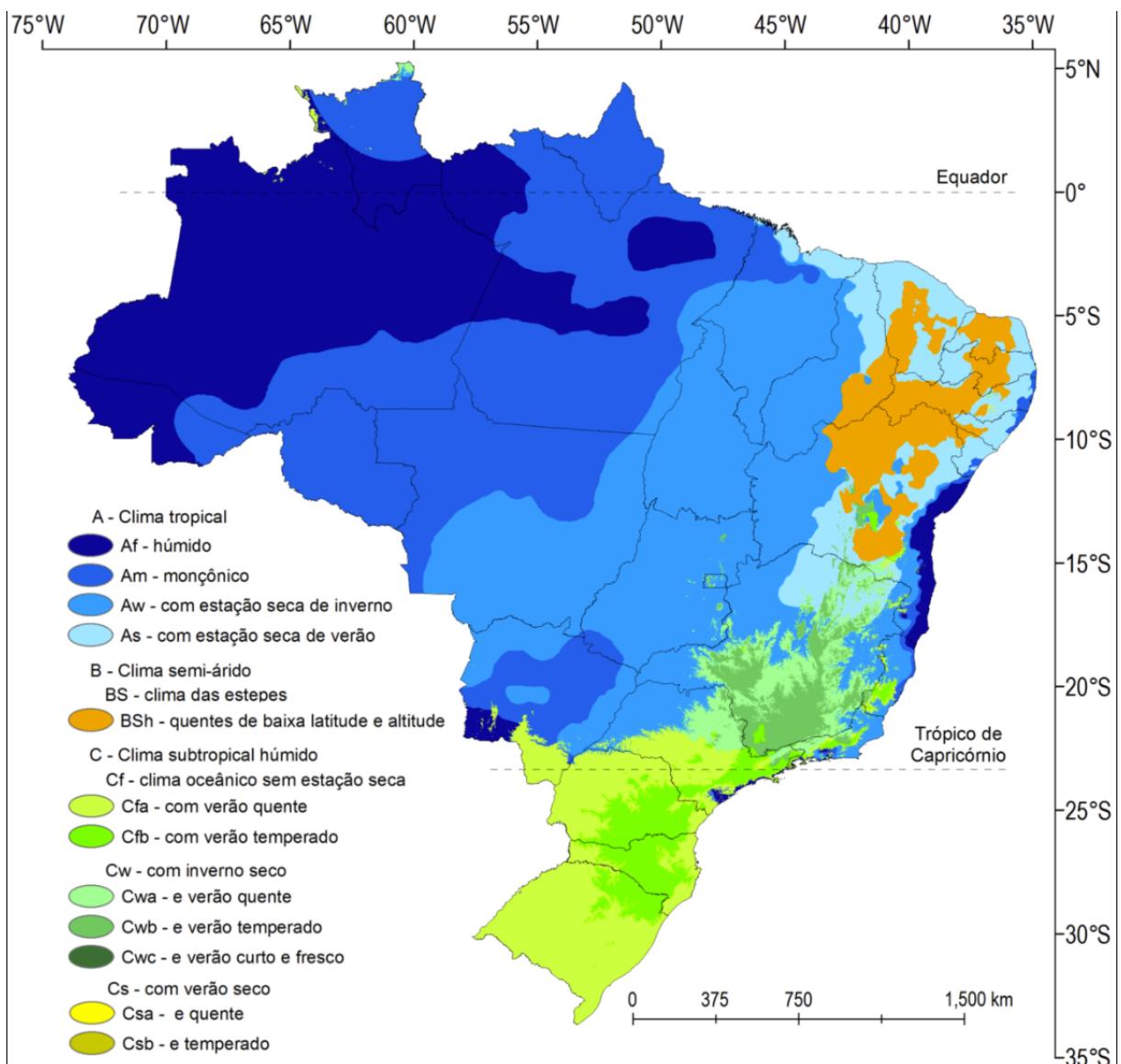


Figura 19 – Mapa de Climas do Brasil por Alvares et al.(7) retirado de (7)