Submission date: 25/11/2019
Resubmission date: 01/05/2020
Resubmission date: 01/07/2020
Camera ready submission date: 01/07/2020

1st round notification: 09/03/2020
2nd round notification: 09/06/2020
3rd round notification: 01/07/2020
Available online: 06/07/2020
Publication date: 31/07/2020

Section: extended version
Extended versions of papers presented at the XV Brazilian Symposium on Information Systems (SBSI 2019)

Minerando Regras de Associação de Multirrelação na Web de Dados

Mining Multirelation Association Rules on the Web of Data

Felipe Alves de Oliveira[1,2], Guilherme dos Santos Villote[1], Raquel Lopes Costa[3], Ronaldo R. Goldschmidt[1], Maria Cláudia Cavalcanti[1]

[1]Instituto Militar de Engenharia (IME), Urca, Rio de Janeiro – Brasil

[2]Instituto de Pesquisas Jardim Botânico do Rio de Janeiro, Jardim Botânico, Rio de Janeiro – Brasil.

[3]Instituto Nacional do Câncer (INCA), Centro, Rio de Janeiro – Brasil

felipealves@jbrj.gov.br, guilherme.villote@ime.eb.br, quelopes@lncc.br,{ronaldo.rgold, yoko}@ime.eb.br

Abstract. *The Web of Data is a growing and relevant data source that contains information distributed in interconnected datasets. Most data mining algorithms was designed to analyze a single dataset at a time and, hence, cannot explore the connections between the datasets in the Web of Data. To overcome this limitation, the present work proposes MRAR+, a graph mining method that searches for multirelation association rules in order to identify new and useful knowledge involving resources of multiple datasets connected to the Web of Data. To illustrate MRAR+'s feasibility, this paper reports on two experiments where the proposed method mined different datasets interconnected in the Web of Data and produced new and useful rules for the users.*

Keywords. *Multirelation Association Rule Mining; Web of Data; Graph Mining; Data Mining*

Resumo. *A Web de Dados é uma relevante e crescente fonte de dados que contém informações distribuídas em diferentes datasets interligados. A maioria dos algoritmos de mineração de dados foi projetada para analisar um único dataset por vez, e, consequentemente, não consegue explorar as conexões entre datasets da Web de Dados. Para suprir essa lacuna, este trabalho propõe o MRAR+, um algoritmo de mineração de grafos que busca por regras de associação multirrelação a fim de identificar conhecimentos novos que envolvam recursos de múltiplos datasets da Web de Dados. O MRAR+ foi aplicado com sucesso em dois experimentos e produziu regras novas e úteis para os usuários, ilustrando a sua viabilidade para minerar diferentes datasets interligados na Web de Dados.*

Palavras-Chave. *Mineração de regras de associação de multirrelação; Web de Dados; Mineração de Grafos; Mineração de Dados.*

Cite as: Oliveira, F. A.., Villote, G. S., Costa, R. L., Goldschmidt, R. R. & Cavalcanti, M. C. (2020). Mining Multirelation Association Rules on the Web of Data (Minerando Regras de Associação de Multirrelação na Web de Dados). iSys: Revista Brasileira de Sistemas de Informação (Brazilian Journal of Information Systems), 13(4), 77-100. DOI: 10.5753/isys.2020.830

1. Introdução

Atualmente a Web possui conjuntos de dados (*datasets*) extremamente volumosos, dinâmicos e diversificados de informações. Tais conjuntos são resultado, em grande parte, da facilidade com que as pessoas são capazes de gerar conteúdo através de computadores ou de seus dispositivos móveis e compartilhá-los na internet. Embora exista toda essa riqueza e heterogeneidade de dados disponíveis na rede, a Web foi projetada inicialmente para ser visualizada e interpretada por seres humanos. Poucos dados da Web possuem representação adequada para que sua semântica (ie., seu significado) possa ser processada automaticamente por máquinas. Além disso, a grande maioria deles sequer encontra-se interligada ou apresenta integração consistente.

Dados interligados é uma premissa para a Web de Dados (ou Web Semântica) que foi proposta por Berners-Lee, o mesmo criador da Web, em 2001 [Bizer et al. 2009, Berners-Lee et al. 2001]. A ideia da Web de Dados é ser um espaço global de dados e seus respectivos significados, representados de forma adequada à compreensão tanto por humanos quanto por computadores. Com os dados conectados e representados de modo formal, por meio de grafos, como sugere Berners-Lee, é possível fazer consultas complexas e inferências. Iniciativas como o projeto Linked Open Data (LOD)[1] têm contribuído significativamente para a construção da Web de Dados. Esse projeto estabelece princípios para a criação do que se chamou de nuvem de dados abertos e interligados (LOD Cloud), que atualmente contém cerca de 10 mil datasets[2]. Apesar dos avanços da Web de Dados, ainda permanece o desafio de processar e analisar eficientemente tal volume de dados em grafo.

Além disso, vale mencionar que a nuvem do LOD cresceu devido ao requisito de se estabelecer ligações entre datasets, e isso ocorreu através da criação de ligações do tipo identidade (*owl:sameAs links*), onde um item de dado é declarado como tendo o mesmo significado que outro pertencente a outro *dataset* [Raad et al. 2018]. Ainda é um desafio estabelecer associações de diferentes tipos entre *datasets* distintos.

Recentemente, estratégias de mineração de dados em estruturas de grafos têm sido propostas na literatura [Barati et al. 2017, Barati 2019, Hendrickx et al. 2015, Elseidy et al. 2014, Ait-Mlouk et al. 2019, Ramezani et al. 2014]. Embora essa perspectiva seja inovadora, os trabalhos atuais fazem uso de algoritmos de mineração sobre grafos explorando apenas um *dataset*, o que não contribui para resolver o problema da integração na Web de Dados.

Dessa forma, o objetivo deste trabalho é apresentar um método que viabilize a análise de múltiplos *datasets* da Web de Dados, de modo a contribuir para a integração dos mesmos. Denominado MRAR+, o método proposto procura enriquecer o conteúdo de um dado *dataset* arbitrário d, a partir de outros *datasets* da Web de Dados que tenham potencial para incorporar conhecimentos novos e úteis em d. O MRAR+ utiliza técnicas de mineração de regras de associação em grafos para identificar itens de dado a partir dos quais d será enriquecido. Os experimentos apresentados neste trabalho ilustram o funcionamento do MRAR+ e seu potencial como alternativa para viabilizar a análise de

[1] https://lod-cloud.net/

[2] http://lodstats.aksw.org/

dados no cenário da Web de Dados, e sugerir novas associações entre itens de dado de diferentes *datasets* desse cenário.

Este trabalho é uma versão estendida de trabalhos anteriores [de Oliveira et al. 2019] [Oliveira et al. 2017], que apresenta um novo experimento baseado em dados sintéticos, mostrando mais detalhes sobre a interface e trazendo resultados promissores sobre a escalabilidade da ferramenta implementada.

Este artigo foi dividido em oito seções. Esta primeira seção faz uma introdução breve sobre Web de Dados e seus desafios dentro do contexto da nossa proposta. Na segunda seção são levantados os conceitos básicos importantes relacionados com a proposta do trabalho. Já os trabalhos relacionados são apresentados na Seção 3. Em seguida apresenta-se uma formalização da mineração de regras de associação de multirrelação (Seção 4). A Seção 5 apresenta a solução proposta seguida da implementação do algoritmo MRAR+ (Seção 6) e dos experimentos (Seção 7). A Seção 8 aponta as principais contribuições deste trabalho e perspectivas de trabalhos futuros.

2. Conceitos Básicos

2.1. Web de Dados

A Web de Dados pode ser expressa como uma rede conectada de informações. Os nós dessa rede estão semanticamente ligados, formando um grande grafo global, com informações advindas de várias fontes diferentes ao redor do planeta [Bizer et al. 2008, Bizer et al. 2009]. Essa rede é composta por conjuntos de dados expressos segundo o modelo de dados conhecido como RDF (Resource Description Framework). Nesse modelo, cada conjunto versa sobre um domínio de conhecimento (por exemplo, termos médicos, localidade geográfica e música) e é possível representar as informações (ou recursos Web) por meio de triplas que são compostas por: Sujeito, Predicado e Objeto, sendo essas organizadas como um grafo direcionado. O Sujeito é o recurso descrito; o objeto pode ser um valor literal ou um recurso relacionado ao sujeito; e o predicado, também chamado de propriedade, indica a relação que existe entre o sujeito e o objeto. Cada elemento das triplas é identificado por uma URI (Uniform Resource Identifier) [Tavares et al. 2015]. Como exemplo de *dataset*, temos o *DBpedia*[3] cuja versão em inglês descreve aproximadamente 4,58 milhões de itens (pessoas, lugares, músicas, filmes, instituições, etc.), dos quais 4,22 milhões classificam-se em uma ontologia consistente[4].

Os *datasets* na Web de Dados permitem acesso ao seu conteúdo por meio de navegação (Web crawling), RDF *dump* ou via linguagem de consulta conhecida como SPARQL. Como esses *datasets* estão interligados, o usuário pode iniciar sua busca em um *dataset* e, logo em seguida, mover-se através dos recursos que os interligam, podendo alcançar intermináveis *datasets* [Pickler 2007, Vieira et al. 2012]. Por exemplo, ao navegar pelo *dataset* do *DBpedia*, na página (*site*) que descreve o recurso que representa a cidade do Rio de Janeiro, encontra-se referência à seguinte tripla presente naquele *dataset*:

[3]http://dbpedia.org/
[4]http://wiki.dbpedia.org/about/

```
http://dbpedia.org/resource/Rio_de_Janeiro_(city)
http://www.w3.org/2002/07/owl#sameAs
http://www.wikidata.org/entity/Q8678
```

Nessa tripla, a propriedade "sameAs" pertence ao vocabulário da linguagem OWL (Web Ontology Language) e indica uma ligação de identidade entre o sujeito e o objeto da tripla. Além disso, observa-se que o objeto da tripla tem uma URI de outro *dataset*, o Wikidata. Assim, ao acionar a URI deste objeto, o usuário passa a navegar pelos dados deste outro *dataset*. No dataset *DBpedia* pode-se encontrar ainda uma outra tripla com o mesmo sujeito, que através da mesma propriedade ("*sameAs*"), leva a um recurso do *dataset* GeoNames (http://sws.geonames.org/3451190/). Nesse caso, ao ativar o objeto da tripla, além de obter informações complementares, obtém-se também um mapa e sua geolocalização.

Os objetos presentes em um *dataset*, que apontam para outros *datasets*, passaremos a denominar "recursos externos".

2.2. Mineração de Regras de associação

O fortalecimento da Web de Dados através de iniciativas como o *dataset DBpedia*, tem permitido a utilização de mecanismos computacionais inteligentes para exploração desses dados na busca por conhecimento em diferentes áreas de aplicação [de Oliveira et al. 2019]. Nesse contexto, técnicas de descoberta de conhecimento em bases de dados (do inglês, *Knowledge-Discovery in Databases* – KDD) podem ser empregadas. KDD é um processo para extrair informações de bases de dados, identificando relações de interesse que normalmente não são observadas pelos especialistas no assunto, podendo ainda auxiliar na validação do conhecimento extraído [Fayyad et al. 1996].

Uma das atividades mais relevantes no KDD é a mineração de dados. A mineração de dados busca encontrar padrões recorrentes e detectar relacionamentos entre variáveis a partir da exploração de grandes quantidades de dados. Para alcançar tal objetivo, utiliza-se de diversos algoritmos, técnicas e tarefas, como a mineração de regras de associação (do inglês, *Association Rules Mining* – ARM) [Goldschmidt et al. 2015].

A ARM consiste em identificar regras de associação frequentes e válidas em um conjunto de dados [Agrawal et al. 1993]. Uma regra de associação *R* é uma implicação da forma $X \rightarrow Y$, onde *X* e *Y* são conjuntos de itens tais que $X \cap Y = \emptyset$. Um item é uma condição que pode assumir valor verdadeiro ou falso em função do registro de dados selecionado. Por exemplo, $R_1 : \{Sexo = M, JogaFutebol = S\} \rightarrow \{Saude = Boa\}$ é uma regra de associação onde $Sexo = M$, $JogaFutebol = S$ e $Saude = Boa$ são itens. Satisfazem a R_1, todos os registros do conjunto de dados que satisfazem (i.e., tornam verdadeiros) esses três itens ao mesmo tempo.

Uma regra de associação $R : X \rightarrow Y$ é dita frequente (resp. válida) se, e somente se, $Sup(R) = |X \cup Y|/|D| \geq$ ***MinSup*** (resp. $Conf(R) = |X \cup Y|/|X| \geq$ ***MinConf***), onde $X \cup Y$ é o conjunto de registros que satisfazem os itens em *X* e *Y* simultaneamente; $Sup(R)$ e $Conf(R)$ são, respectivamente, o suporte e a confiança de R; $|D|$ representa a quantidade total de registros de dados disponíveis no conjunto de dados D; e ***MinSup*** e ***MinConf*** são parâmetros definidos pelo usuário. Considerando a regra R_1 exemplificada

anteriormente, se o conjunto de dados tem 100 registros no total, dos quais 50 satisfazem simultaneamente aos itens Sexo=M e JogaFutebol=S, mas apenas 45 desses, satisfazem também ao item Saude=Boa, tem-se que $Sup(R_1) = 45\%$ e $Conf(R_1) = 90\%$. Assumindo, por exemplo, ***MinSup*** $= 10\%$ e ***MinConf*** $= 80\%$, R_1 seria considerada uma regra frequente e válida.

Em geral, o processo de ARM ocorre em duas etapas [Agrawal et al. 1993]. A primeira, de maior custo computacional, busca por conjuntos de itens frequentes que ocorrem simultaneamente no conjunto de dados. A segunda consiste em identificar as regras válidas a partir de cada conjunto de itens considerado frequente na etapa anterior.

3. Trabalhos Relacionados

Técnicas de mineração sobre dados estruturados em grafos têm sido um tema de interesse recorrente e largamente encontrado na literatura científica [Rehman et al. 2012]. Esses trabalhos têm objetivos diversos como encontrar padrões de links na Web de dados [Zhang et al. 2012], sumarizar grafos [Sydow et al. 2013], caracterizar o conteúdo de grafos RDF [Basse et al. 2010], entre outros. Destacamos a seguir alguns trabalhos mais relacionados com o presente trabalho, e que têm como objetivo encontrar regras de associação em grafos.

Por exemplo, o trabalho de [Hendrickx et al. 2015] tem como objetivo encontrar interações (arestas) adicionais a partir da ARM entre os rótulos dos nós de um grafo não direcionado. Os itens frequentes foram identificados através da análise sobre esses rótulos. Com isso, são reconhecidos os conjuntos que são, em média, frequentes, mesmo que não estejam exatamente relacionados da mesma maneira. O algoritmo proposto pelos autores descobre regras que permitem afirmar que, se um conjunto de rótulos de nós é encontrado em um grafo, há uma alta probabilidade de que algum outro conjunto de rótulos possa ser encontrado na sua proximidade. Apesar de interessante, este trabalho propõe-se a analisar apenas os rótulos dos nós, mas não leva em consideração a análise sobre a composição das relações.

O trabalho de [Elseidy et al. 2014] propõe uma abordagem para a mineração de subgrafos frequentes em um grafo grande não direcionado. Essa abordagem procura por conjuntos mínimos de casos de forma a satisfazer o nível de frequência e evitar a enumeração custosa de todos os casos. A avaliação dos subgrafos é tratada como um problema de satisfação de restrições (CSP – *Constraint Satisfaction Problem*). A cada iteração, ele resolve o CSP até encontrar o conjunto mínimo que é suficiente para avaliar a frequência do subgrafo, ignorando o conjunto restante. No entanto, esta abordagem não trata grafos dirigidos.

Já o artigo de [Barati et al. 2017, Barati 2019] propõe uma abordagem capaz de revelar regras de associação em grafos dirigidos, no formato RDF, no qual os nós e as arestas do grafo são rotuladas. No grafo RDF, os nós podem ser instâncias ou classes (rótulos). Essa abordagem, chamada de SWARM (Semantic Web Association Rule Mining) pode analisar e gerar automaticamente regras enriquecidas semanticamente (explicitando instâncias e classes). No entanto, as regras geradas não são regras multirrelação, isto é, não evidenciam uma cadeia de caminhos frequentes como faz o trabalho descrito a

seguir.

Um outro trabalho mais recente [Ait-Mlouk et al. 2019] também trabalha sobre grafos RDF. No entanto, seu foco está em extrair regras de associação dos textos encontrados nos atributos do grafo, usando técnicas de processamento de linguagem natural. Dessa forma, esse trabalho não explora os caminhos do grafo, e, como os demais trabalhos, concentra-se em somente um *dataset*.

O trabalho de [Ramezani et al. 2014], a ser detalhado na Seção 4, propõe uma outra abordagem para ARM em grafos RDF, chamada MRAR (*MultiRelation Association Rules*). A proposta do autor é encontrar caminhos frequentes que possam ocorrer em um grafo, considerando os diferentes tipos de relação que formam os caminhos. Assim, as regras de associação multirrelação geradas explicitam a composição das relações dos caminhos do grafo que dão origem às mesmas, trazendo maior entendimento.

A Tabela 1 apresenta uma comparação entre os principais trabalhos relacionados com a proposta do presente artigo. Note que, embora explorem grafos com múltiplos tipos de relação, somente um deles, entre os relacionados, de fato gera regras multirrelação. Outro ponto a destacar é que somente três deles, trabalham sobre grafos direcionados e consideram o formato RDF. Além disso, todos eles analisam apenas um *dataset*, e não exploram as ligações com outros *datasets*, o que poderia favorecer a identificação de novas regras de associação.

A principal contribuição deste trabalho é envolver mais de um *dataset* na mineração de regras de associação de multirrelação. Desenvolvido como uma extensão do algoritmo MRAR, o MRAR+ considera os recursos externos presentes no grafo RDF e permite que novas regras de associação multirrelação sejam geradas.

Tabela 1. Tabela comparativa dos artigos que utilizam mineração de dados sobre grafos.

Trabalhos Relacionados	**Grafo Direcionado**	**Formato RDF**	**Explora Multi relações**	**Gera regras Multi relação**	**Explora *links* entre datasets**
[Hendrickx et al. 2015]	Não	Não	Sim	Não	Não
[Elseidy et al. 2014]	Não	Não	Sim	Não	Não
[Ait-Mlouk et al. 2019]	Sim	Sim	Não	Não	Não
[Barati 2019]	Sim	Sim	Sim	Não	Não
[Ramezani et al. 2014]	Sim	Sim	Sim	Sim	Não
MRAR+	Sim	Sim	Sim	Sim	Sim

4. Mineração de Regras de Associação de Multirrelação

Inspirada nos conceitos apresentados na Subseção 2.2, a Mineração de Regras de Associação de Multirrelação é uma adaptação da tarefa de ARM a fim de identificar regras em grafos dirigidos representados no formato RDF [Ramezani et al. 2014]. Nesse

contexto, a ideia é encontrar caminhos frequentes que possam ocorrer em um grafo e que cheguem em um mesmo recurso a partir de instâncias de determinadas propriedades desse grafo. A seguir encontra-se uma descrição resumida das definições extraídas de [Oliveira et al. 2017], que são a base para a busca por regras de associação multirrelação frequentes e válidas. Os exemplos utilizados para ilustrar as definições apresentadas tomam como referência o grafo da Figura 1.

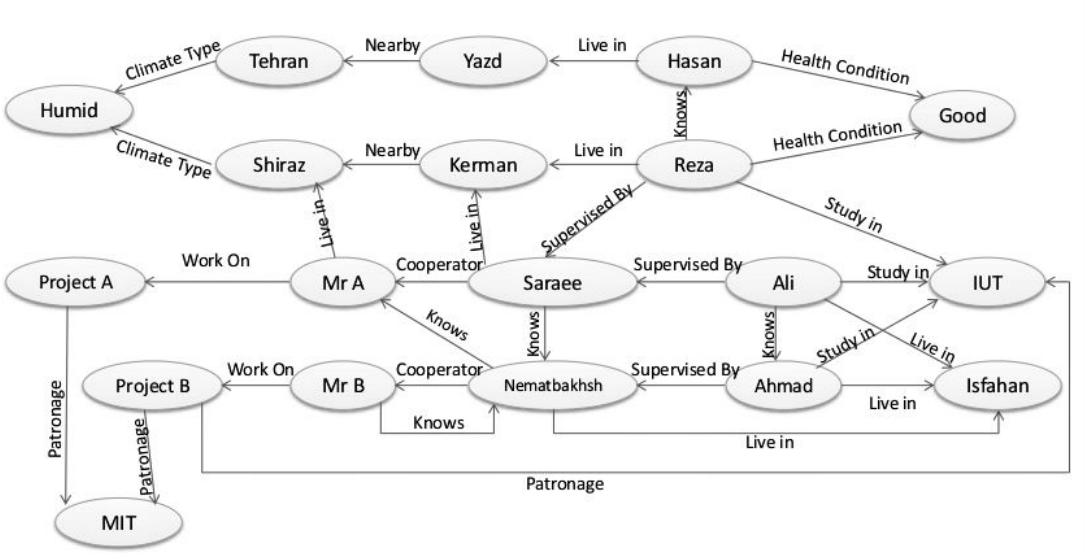

Figura 1. Exemplo de um grafo direcionado com rótulo nas arestas. Extraído de: [Ramezani et al. 2014].

1. Denomina-se caminho $C(x, y)$ em um grafo G a um conjunto ordenado de instâncias das propriedades $p^1, ..., p^k$ que, ao serem aplicadas, levam um recurso x (chamado de origem do caminho) a um recurso y (chamado de destino do caminho). $C(Reza, Humid)$ = Reza $\xrightarrow{Live\_in}$ Kerman $\xrightarrow{Near\_by}$ Shiraz $\xrightarrow{Climate\_Type}$ Humid é um exemplo de caminho entre os recursos *Reza* (origem) e *Humid* (destino).
2. Define-se cadeia $\mathcal{C}_{y,(p^1,...,p^k)}$ de um grafo G como um conjunto de caminhos de G que passam por instâncias das propriedades $p^1, ..., p^k$ e chegam em um recurso y. Por exemplo: os caminhos
   Reza $\xrightarrow{Live\_in}$ Kerman $\xrightarrow{Near\_by}$ Shiraz $\xrightarrow{Climate\_Type}$ Humid, e Hasan $\xrightarrow{Live\_in}$ Yazd $\xrightarrow{Near\_by}$ Tehran $\xrightarrow{Climate\_Type}$ Humid, são elementos da cadeia
   $\mathcal{C}_{Humid,(Live\_in,Near\_by,Climate\_type)}$.
3. Considera-se $\mathcal{I}(C_{y,s})$ a coleção de recursos que são origens em caminhos que pertençam à cadeia $\mathcal{C}_{y,s}$, onde $s = (p^1, ..., p^k)$. No exemplo,
   $\mathcal{I}(C_{Humid,(Live\_in,Near\_by,Climate\_type)})$ = $\{Reza, Hasan\}$.
4. Assim, define-se regra de associação de multirrelação entre duas cadeias $\mathcal{C}_{y,s}$ e $\mathcal{C}_{z,r}$ como sendo uma implicação $\mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$, onde $\mathcal{C}_{y,s} \cap \mathcal{C}_{z,r} = \emptyset$.
   $R_2 : Live\_In(Near\_By(Climate\_Type(Humid))) \rightarrow$
   $Health\_Condition(Good)$ é um exemplo de regra de associação de multirrelação.

Segundo ela, indivíduos que vivem em cidades próximas de cidades que possuem clima úmido são indivíduos que apresentam condição de saúde boa.

5. Por fim, diz-se que uma regra de associação de multirrelação $R : \mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$ é frequente (resp. válida) se, e somente se, $Sup(R) = |\mathcal{I}(C_{y,s}) \cap \mathcal{I}(C_{z,r})|/|V| \geq$ *minsup* (resp. $Conf(R) = |\mathcal{I}(C_{y,s}) \cap \mathcal{I}(C_{z,r})|/|\mathcal{I}(C_{y,s})| \geq$ *minconf*), onde V é o conjunto de recursos (nós) do grafo. No contexto do exemplo de onde a Figura 1 foi extraída, existem ao todo 19 recursos, e foram considerados $MinSup = 10\%$ e $MinConf = 60\%$ [Ramezani et al. 2014]. Assim, como naquele contexto, $Sup(R_2) = 11\%$, $Conf(R_2) = 69\%$, a regra R_2 indicada no item anterior é considerada frequente e válida.

As definições acima foram concebidas com base no trabalho que apresenta o MRAR [Ramezani et al. 2014], um dos primeiros algoritmos de busca por regras de associação de multirrelação frequentes e válidas observados na literatura da área de mineração de dados. Por este motivo, serviu de base para a implementação da abordagem proposta no presente artigo.

5. Solução Proposta

Como foi descrito anteriormente, o problema que deseja-se tratar neste trabalho caracteriza-se pela dificuldade de realizar análises em grandes volumes dados, organizados na forma de grafo, pois há vasta diversidade de tipos de nós e de relações para que se possa extrair informação útil. Esses dados estão dispostos em *datasets* interligados na Web de Dados, em formato RDF. Atualmente, sabe-se que é inviável tratar o conjunto completo destes *datasets*, devido à quantidade de dados envolvida. Além disso, há uma necessidade de encontrar novos tipos de relações entre *datasets* distintos, enriquecendo dessa forma as ligações na Web de Dados.

Assim, o objetivo desse estudo é apresentar uma abordagem para esse problema, através da ampliação de um determinado *dataset* (*dataset* alvo) com informações complementares, de maneira controlada e temporária. A ideia é enriquecer com informações de outros *datasets* (*datasets* externos), e com essa ampliação, encontrar novas regras de associação de multirrelação, aumentando o conhecimento sobre os dados analisados. Adicionalmente, a partir de tais regras é possível sugerir novas relações entre recursos dos distintos datasets.

Por exemplo, no item (4) da seção anterior, a regra R_2 sugere uma nova relação onde clima úmido é propício à saúde boa. Supondo que a informação sobre o clima das localidades estivesse em um *dataset*, e que a informação sobre a saúde e localidade das pessoas estivesse em outro *dataset*, não seria possível inferir tal relação analisando somente um dos *datasets*. No entanto, isso torna-se possível a partir da mineração de um *dataset* ampliado, isto é, que integre ambas as informações.

Nossa proposta assume que temos um *dataset* alvo, que já inclui recursos apontando para *datasets* externos, através de ligações *owl:sameAs*. A hipótese é que, ao realizar a mineração de regras de associação de multirrelação sobre o *dataset* alvo, entre os recursos que dão suporte às regras encontradas, seja possível identificar recursos externos, que apontam para *datasets* externos. A partir dos *datasets* selecionados (um subconjunto

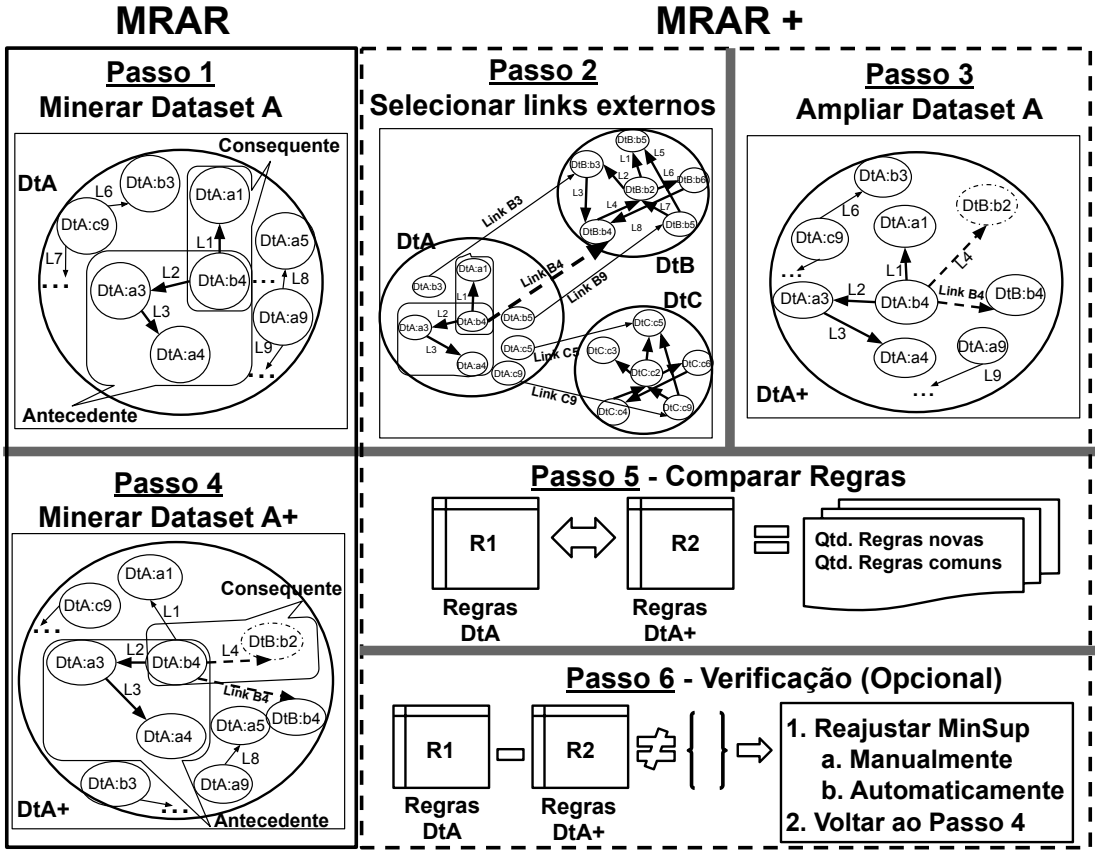

Figura 2. Visão geral ilustrando cada um dos passos aplicados a proposta. Os quadros pontilhados demonstram os passos dados pelo algoritmo proposto (*MRAR+*) e os quadros simples se referem aos passos dados pelo algoritmo existente (MRAR).

dos *datasets* externos), é possível enriquecer o *dataset* alvo com informações úteis (novas triplas extraídas dos *datasets* selecionados), ampliando assim seu conhecimento. Desta maneira, ao minerar o *dataset* enriquecido, possibilita-se encontrar novas regras e evita-se ter que minerar o *dataset* alvo e todos os *datasets* externos por completo.

Além disso, a integração com um *dataset* externo também não é completa. A ideia é restringir/selecionar apenas os dados (recursos) de maior relevância, bastando para isso utilizar como critério, a seleção dos recursos do grafo que servem como suporte para as regras encontradas. Com esse critério, mesmo que um *dataset* possua muitos links externos, são utilizados apenas os que foram selecionados, o que viabiliza a análise em *datasets* com muitas conexões.

A Figura 2 mostra todos os passos da abordagem proposta aplicados para o processo de ampliação de conhecimento de um *dataset* hipotético chamado DtA. Inicia-se com a mineração do *dataset* alvo (DtA, passo 1), passando pela seleção de recursos externos (passo 2). Uma vez identificados os recursos externos, passa-se, então, para o processo de ampliação do conhecimento existente no DtA (passo 3), com as informações encontradas no *dataset* externo selecionado (DtB), gerando o *dataset DtA+*. Em seguida

(passo 4), minera-se esse novo conjunto de dados (*DtA+*) para encontrar novas regras.

Na implementação do protótipo, o algoritmo MRAR foi utilizado nos passos 1 e 4. Para contemplar os passos 2 e 3, o algoritmo *MRAR+* foi desenvolvido. É importante destacar que o *MRAR+* também é responsável por dar início à execução do passo 4, passando os novos parâmetros para o MRAR.

O algoritmo *MRAR+* foi escrito de forma amigável (*user-friendly*), permitindo que o usuário faça ajustes durante alguns passos de sua execução. Dessa forma, no passo 5, o usuário tem a chance de comparar o conjunto de regras geradas sobre o *dataset DtA* com o conjunto gerado pela mineração sobre o *dataset DtA+*. Já no passo 6, o usuário tem a opção de reajustar o valor de *MinSup* para permitir que o valor utilizado seja suficiente tanto para a geração das regras novas quanto das antigas, encontradas na primeira execução do MRAR. Neste caso, é preciso executar os passos 4 e 5 novamente. A seguir, serão apresentados todos os passos dessa visão geral com esquemas ilustrativos para facilitar o entendimento e demonstrar a aplicação de cada passo.

No passo 1, ao aplicar a mineração de dados com o algoritmo MRAR, é possível encontrar regras com cadeias do tipo $\mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$ (conforme definição 4 da Seção 2). As regras cujo valor de suporte não atenderem ao valor de suporte mínimo, serão eliminadas nesse passo. Na ilustração da Figura 2 (passo 1), temos que apenas uma regra foi encontrada: L2(L3(*DtA:a4*)) $\rightarrow$ L1(*DtA:a1*). Para cada regra encontrada, um conjunto de recursos é selecionado para servir de entrada para o passo seguinte. Cada conjunto representa os recursos que dão suporte à regra, que para a regra do exemplo, é definido como $\mathcal{I}(C_{DtA:a4,(L2,L3)}) \cap \mathcal{I}(C_{DtA:a1,(L1)})$ (conforme a definição (5) da Seção 2).

No passo seguinte, o algoritmo *MRAR+* seleciona somente os recursos que são externos nos conjuntos gerados no passo 1. Mesmo que o *dataset* alvo possua outros recursos externos, a ideia aqui é evitar trabalhar com todos eles, selecionando apenas os recursos que têm potencial para a descoberta de novas regras. Já que tais recursos foram suporte para encontrar regras no *dataset* alvo (sem expansão), esses têm maior chance de se tornar recursos que possam dar suporte para novas regras no *dataset* alvo estendido. No exemplo da Figura 2(1), é possível visualizar o recurso *DtA:b4* como um dos recursos que dão suporte à regra selecionada ao aplicar o MRAR no *DtA*, no passo 1. Também é possível observar que o recurso *DtA:b4* do *dataset DtA* está ligado ao recurso externo *DtB:b4* do *dataset DtB*, como mostra a seta pontilhada com o nome "link b4" na Figura 2(2). Como mencionado na Seção 2, na maioria dos casos, essas ligações são do tipo *owl:sameAs*. Assim, ao final deste passo, somente o recurso *DtB:b4* foi selecionado para o passo seguinte.

No passo 3, o *MRAR+* busca, nos *datasets* apontados pelos recursos externos selecionados no passo anterior, as novas triplas envolvendo esses recursos e seus vizinhos. Neste exemplo, limitou-se à vizinhança apenas de caminhos de comprimento 1 (apenas 1 aresta), e apenas um *dataset* externo. As triplas encontradas no *dataset* externo têm como sujeito o recurso externo. Assumindo que essas triplas foram obtidas a partir de ligações do tipo *owl:sameAs*, substituímos o recurso externo pelo recurso local do *dataset* alvo (*DtA*), ampliando dessa forma suas informações. No exemplo da Figura 2(3), nota-se que a tripla (*DtA:b4* $\xrightarrow{L4}$ *DtB:b2*) foi adicionada ao DtA. A junção entre os dados do

dataset DtA e de um subconjunto dos dados extraídos do *dataset DtB* possibilita a criação de um novo conjunto de dados, que foi chamado de *dataset DtA+*. Vale ressaltar que este *dataset* ampliado existirá temporariamente, pois tem como objetivo apenas servir à análise realizada no passo seguinte.

No passo 4, o algoritmo MRAR é executado novamente, chamado a partir do algoritmo *MRAR+*, agora selecionando o novo *dataset* criado (*DtA+*) como entrada. Na Figura 2(4), nota-se que uma nova regra foi encontrada, essa fazendo uso dos nós e arestas trazidos do *dataset* externo (*DtB*). Nesse exemplo, a regra L2(L3(DtA:a4)) $\rightarrow$ L4(DtB:b2) não poderia ser encontrada se apenas os dados do *DtA* fossem utilizados.

No passo 5 da Figura 2, as regras obtidas com a mineração sobre os dados originais (R1 do *DtA*) são comparadas com as regras encontradas após o processo de extensão (R2 do *DtA+*). Essa comparação gera algumas estatísticas, demonstrando o total de regras que são novas e o total de regras comuns às duas execuções.

Por fim, no passo 6 da Figura 2, o usuário tem a opção de verificar se todas as regras (R1) encontradas durante a primeira execução do algoritmo, sobre os dados do DtA, fazem parte do conjunto de regras (R2), obtidos após a análise do *dataset* estendido (DtA+). Funciona da seguinte maneira, se R1 - R2 $\neq \{\}$ então o usuário pode reajustar o *MinSup* manualmente ou selecionar a opção para o próprio *MRAR+* calcular o valor de suporte mínimo, tal que R1 esteja contido em R2. Com esse ajuste, os passos 4 e 5 são executados novamente.

O pseudo-código do algoritmo *MRAR+* pode ser encontrado em [de Oliveira et al. 2019]. Vale notar que, embora o algoritmo especificado seja genérico e permita que quaisquer *datasets* externos sejam analisados, somente os recursos externos que dão suporte a alguma regra inicialmente encontrada sobre o *dataset* alvo, serão investigados, reduzindo, dessa forma, o número de *datasets* a investigar.

6. Implementação do MRAR+

A implementação do algoritmo *MRAR+* foi feita utilizando como linguagem de programação o PHP[5] (em sua versão mais recente, o PHP 7.3). O código fonte do algoritmo está disponível para consulta e *download* no repositório do GitHub [6]. Como parte do processo, o algoritmo MRAR também foi implementado e seus resultados servem de entrada para o *MRAR+*.

A interface inicial do algoritmo implementado disponibiliza alguns campos de preenchimento obrigatório, para receber os parâmetros essenciais que são utilizados do processo de descoberta de regras de associação de multirrelação. São eles: *MinLevel*, *MaxLevel*, *MinSup* e *MinConf*. Também é necessário selecionar um *dataset* que será analisado (*DS*) e informar o *endpoint*, que permita consultas na linguagem SPARQL, a ser acessado para consumir as informações dos recursos externos. Logo em seguida, é possível observar o campo "Predicates to external resources", que apresenta os predicados mapeados para serem utilizados para buscar as informações nos *datasets* externos. É

[5]https://www.php.net/

[6]https://github.com/feliperj629/MRAR_plus/

importante destacar que esses valores são lidos de um arquivo interno que foi informado pelo usuário e que novos predicados podem ser incluídos sempre que for necessário. A Figura 3 mostra a tela que permite a inclusão dessas informações iniciais, que são essenciais para execução completa do *MRAR+*.

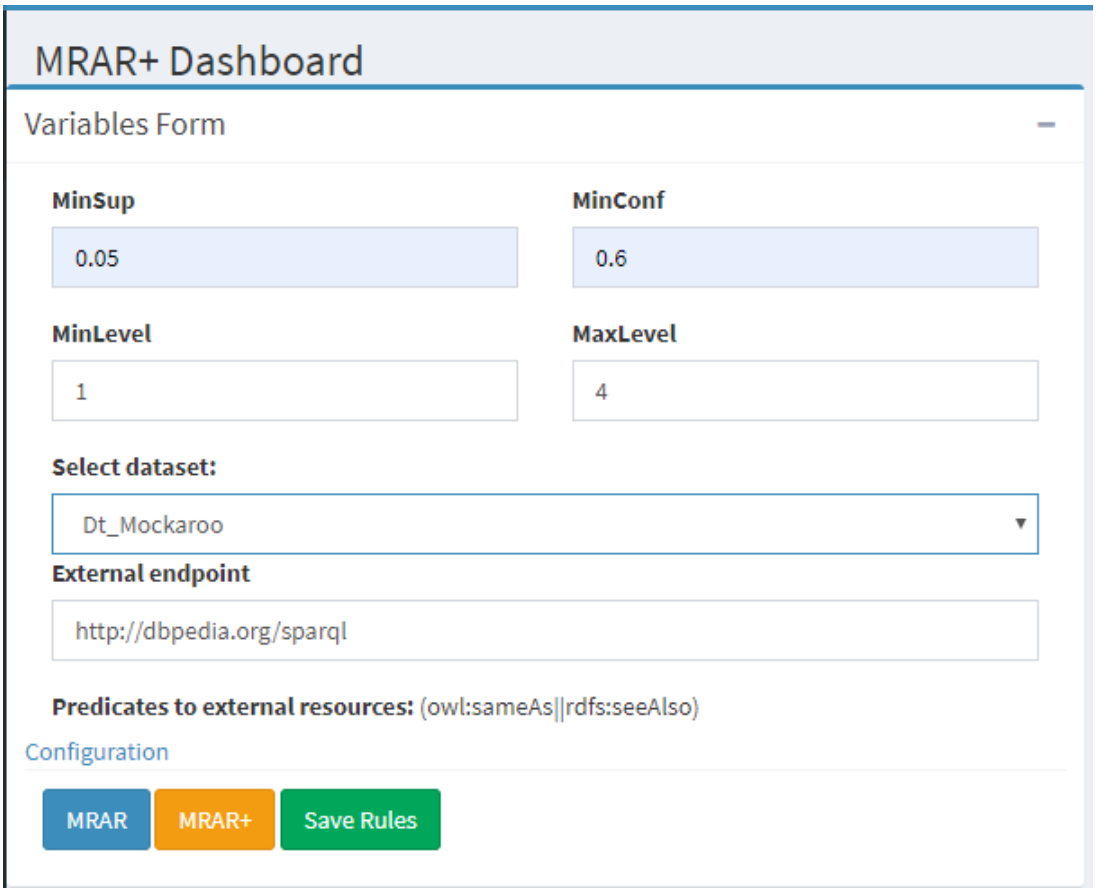

Figura 3. Tela de configuração para as variáveis de entrada do algoritmo *MRAR+*.

A tela do protótipo foi desenvolvida para permitir que o usuário possa ter a escolha de fazer a mineração de dados apenas com o algoritmo MRAR ou de fazer a mineração estendida com o algoritmo *MRAR+*. Funciona da seguinte maneira: para executar somente o MRAR, primeiro é necessário preencher as informações iniciais básicas, como foi detalhado anteriormente. Após o preenchimento, basta clicar no botão MRAR, como é visto na Figura 3. Entretanto, para executar o algoritmo *MRAR+*, além das informações básicas, é preciso informar no campo "External Endpoint" o endereço do *endpoint* que será acessado para buscar as informações externas. Após isso, basta clicar no botão *MRAR+*.

Ao final da execução do *MRAR+*, um novo *dataset* é criado contendo o conjunto de dados do *dataset* original somados aos recursos e arestas que foram capturados do *dataset* externo, além de um cabeçalho contendo as informação básicas usadas para sua criação. Esse novo *dataset* será exibido junto à lista de seleção dos *datasets*, com o nome do *dataset* original mais a data de sua criação. Após a criação do novo conjunto de dados, basta preencher as informações iniciais, selecionar o novo *dataset* na lista e clicar no

botão MRAR, para realizar a mineração sobre o *dataset* ampliado.

Sabendo que acrescentar novos itens ao *dataset* original implicará em alteração do suporte, com o qual as primeiras regras foram geradas, foi necessário disponibilizar uma opção para auxiliar o usuário na hora de definir o valor de suporte mínimo. Visando garantir que as regras geradas, após a análise do primeiro *dataset*, pertençam ao segundo conjunto de regras, gerado ao analisar o *dataset* ampliado, assim como foi demonstrado no passo 6 da solução proposta, Seção 5. Para fazer esse uso, basta clicar no link do *dropdown* "*Configuration*", visto na Figura 3, e marcar a opção "*Apply the best support*". Ao iniciar a execução do algoritmo, um novo valor de suporte mínimo será calculado, visando utilizar um que, além de permitir a geração das novas regras, possibilite também identificar as regras geradas com o *dataset* original, antes da sua ampliação.

Assim, objetivando garantir que o suporte escolhido seja adequado, a fim de permitir que as mesmas regras sejam geradas ao analisar os dois *datasets* (original e ampliado), a fórmula $X = S * D/D'$ foi aplicada. Onde X é o novo valor de suporte que se deseja encontrar, D' é a quantidade de nós existentes no novo *dataset*, S é o suporte que foi utilizado originalmente para analisar o *dataset* alvo e D é a quantidade de nós do grafo original. Por exemplo, em uma execução do algoritmo para um grafo com 19 nós, utilizando o suporte mínimo igual a 0.1, se após a extensão, o novo *dataset* criado passa a ter 25 nós, o cálculo é feito da seguinte forma: $X = 0.1 * 19/25$, $X = 1.9/25$, logo, $X = 0.076$, ou seja, o valor de suporte mínimo que melhor se aplica para possibilitar tanto a geração das regras originais quanto as novas será 0.076.

Rules

Row	Ant.	Cons.	Sup.	Conf.	Lift	Conv.
1	11	1	0.13	1.00	0.77	0.925
2	1	11	0.13	1.00	0.77	0.925
3	21	3	0.04	1.00	1.67	1.000
4	13	4	0.05	1.00	1.11	0.968
5	17	6	0.03	0.75	0.75	0.003
6	15	22	0.05	1.00	1.11	0.968
7	24	20	0.05	0.71	0.89	0.002
8	2||11	1	0.03	1.00	0.77	0.925
9	1||2	11	0.03	1.00	0.77	0.925

Figura 4. Tela de visualização das regras geradas, em formato numérico.

O sistema também mostra o tempo gasto na execução, o quanto de memória foi usada, além de abrir duas formas de visualização das regras que foram geradas. A pri-

meira forma de visualização de regras, vista na Figura 4, mostra os antecedentes e consequentes em formato numérico, fazendo referências a cada cadeia de relações que foram geradas, além dos respectivos valores de suporte, confiança, *lift* e convicção de cada regra gerada. Já a segunda, vista na Figura 5, apresenta os valores dos antecedentes e consequentes das regras em formato de texto, proporcionando, assim, melhor entendimento e visualização das regras. A interface permite ainda a filtragem das regras, através das caixas de texto *Antecedent* e *Consequent*, no topo da tela. Para fazer isso, basta preencher com uma expressão, que a ferramenta filtra automaticamente, e somente as regras que contemplam tal expressão serão apresentadas. Essa funcionalidade revelou-se bastante útil para a análise dos resultados [Oliveira et al. 2017].

Formatted Rules

Row	Antecedent	Consequent	Sup.	Conf	Lift	Conv.
1	Live_In (Niterói) ➔	Live_In (Climate_Type (Hot))	0.13	1.00	0.77	0.925
2	Live_In (Climate_Type (Hot)) ➔	Live_In (Niterói)	0.13	1.00	0.77	0.925
3	Supervised_By (Anthony Holmes) ➔	Study_In (Razi University)	0.04	1.00	1.67	1.000
4	Supervised_By (Janet Evans) ➔	Study_In (Universidad Justo Sierra)	0.05	1.00	1.11	0.968
5	Supervised_By (Daniel Ross) ➔	Study_In (Universidad Galileo)	0.03	0.75	0.75	0.003
6	Supervised_By (Annie Turner) ➔	Study_In (Universitas Methodist Indonesia)	0.05	1.00	1.11	0.968

Figura 5. Tela de visualização das regras geradas, em formato de texto.

Após a execução do algoritmo, é possível salvar as regras que foram geradas. Na tela, Figura 3, é exibido o botão "*Save Rules*". Ao selecioná-lo, o sistema captura as regras que estão sendo exibidas na tela e salva-as em um diretório.

Com as regras salvas, o usuário tem a opção de compará-las. Para tanto, basta acessar o menu lateral, "*Dashboard*", e ir na opção "*Compare Rules*". Uma nova janela será carregada, trazendo duas opções de seleção. Na primeira, é preciso escolher um dos conjuntos de regras salvas, geradas com base nos *datasets* originais e, na segunda, é necessário selecionar o conjunto de regras geradas com o *dataset* estendido. Ao acionar o botão "*Compare Rules*", a página será atualizada com as informações referentes à comparação feita, mostrando o número de regras que foram geradas com os dados do *dataset* original e do *dataset* estendido. As seguintes estatísticas: regras novas, regras comuns e regras que não foram geradas são exibidas após a comparação, como é possível ver na Figura 6. Além disso, uma tabela contendo somente as regras que são novas é exibida, no final da página, para a comparação feita entre os dados selecionados.

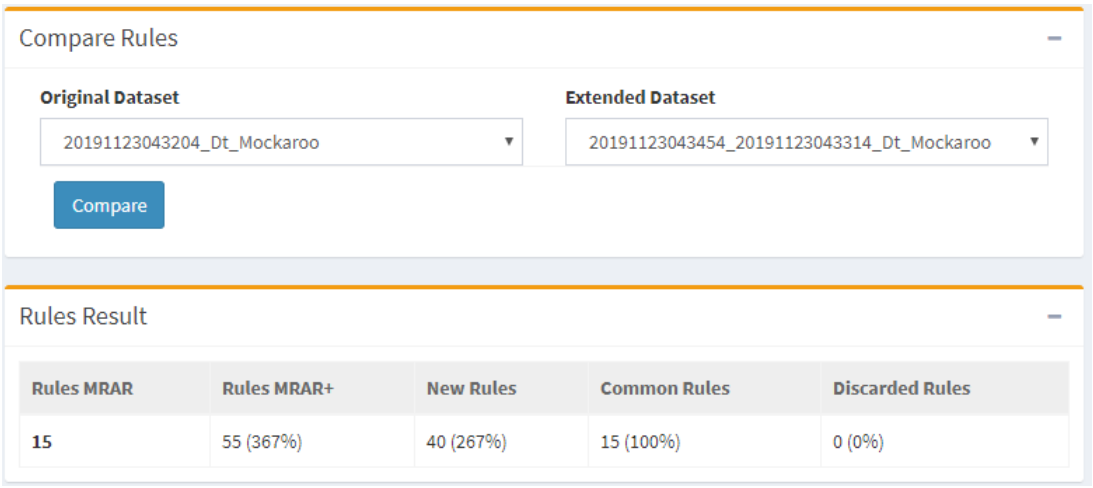

Figura 6. Tela de comparação das regras geradas pelo algoritmo MRAR e *MRAR+*.

A implementação do MRAR+ é um protótipo e ainda apresenta limitações. Uma delas é tratar apenas relações do tipo *sameAs* entre os *datasets*. Outra limitação é a necessidade de atribuir valores muito baixos para o suporte mínimo, de modo a encontrar novas regras.

7. Experimentos

Nesta seção, serão apresentados dois experimentos realizados, a fim de avaliar a abordagem proposta. O experimento 1 teve como objetivo mostrar a funcionalidade do *MRAR+*, tomando como alvo um *dataset* de pequeno porte, com dados conhecidos, de modo a facilitar a análise dos resultados.

No experimento 2, com o objetivo de avaliar a escalabilidade do MRAR+, foi construído um *dataset* alvo de maior porte usando dados sintéticos, e este passou a incluir recursos que apontam para *datasets* já existentes na Web de Dados, mais especificamente, para o *dataset DBpedia*, mostrando dessa forma a funcionalidade completa do *MRAR+*. Outros experimentos realizados foram relatados em [Oliveira 2018].

7.1. Experimento 1 (*dataset* IME - Esportes)

No primeiro experimento, utilizamos como alvo o *dataset* do Instituto Militar de Engenharia (IME), chamado de (DtIME), visto na Figura 7, composto por 36 nós e 91 arestas. Este *dataset* é semelhante ao que foi usado em [Ramezani et al. 2014] (Figura 1), e contém nós dos seguintes tipos: alunos, professores, instituições e cidades. As arestas também são de tipos diferentes: estuda_em (*studies_in*), trabalha_para (*works_on*), orientado_por (*supervised_by*), e mora_em (*lives_in*). Um outro *dataset* foi preparado para ser usado como *dataset* externo, e foi chamado de DtEsportes, contendo um total de 32 nós e 34 arestas. Alguns dos alunos do DtIME são equivalentes aos alunos presentes no *dataset* DtEsportes, onde há informações sobre os esportes que os alunos praticam, como mostra a Figura 7.

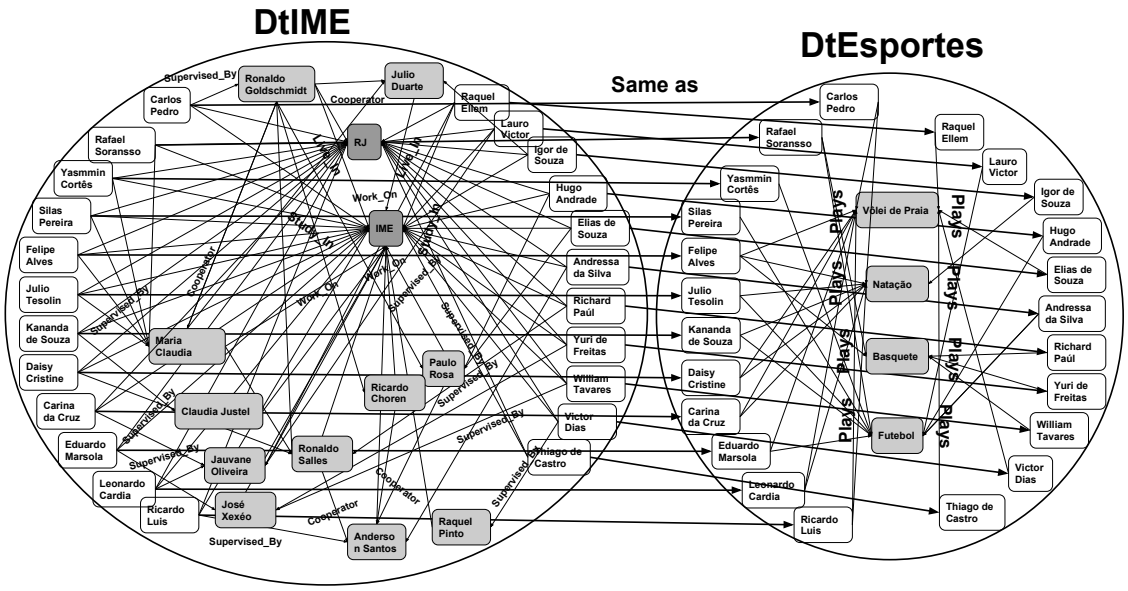

Figura 7. *Datasets* demonstrando associações externas através da relação *Same as*. O *dataset* DtIME apresenta informações dos professores e alunos, juntamente com a localidade onde vivem e a instituição onde trabalham ou estudam. Já o DtEsportes, apresenta os esportes que cada aluno pratica.

Os dados do DtIME e do DtEsportes estão representados em forma de triplas (sujeito, predicado e objeto), no modelo RDF. Algumas URIs dos recursos (nós) existentes em DtIME apontam para recursos do DtEsportes, através da relação *Same_as*. Esses recursos do *dataset* externo, por sua vez, possuem informações complementares sobre os esportes praticados. Exemplos de associação (Same_as) com recursos externos podem ser vistos na Figura 7.

Inicialmente, no passo 1, o algoritmo MRAR é executado sobre o DtIME, com a seguinte configuração de parâmetros: *MinSup* = 10%, *MinConf* = 70%, *MinLevel* = 1 e *MaxLevel* = 4. Como resultado foram encontradas 297 regras.

Em seguida, inicia-se o algoritmo *MRAR+*, que é executado com a mesma configuração de parâmetros. Com base nas regras encontradas no passo 1 e seus recursos (nós) de suporte, o algoritmo *MRAR+* identifica os recursos externos (passo 2), indicando qual *dataset* externo acessar, o que viabiliza o processo de extensão dos dados (passo 3). Neste passo, o *dataset* alvo (DtIME) é estendido (DtIME+), e passa a ter um total de 40 nós e 126 arestas, o que representa um aumento de 11% em números de nós e 38% em relação ao número de arestas, comparado ao *dataset* original. É importante destacar que esse processo permitiu identificar os recursos externos com potencial de gerar novas regras, uma vez que os nós selecionados estavam diretamente ligados aos recursos que deram suporte às primeiras regras. Sendo assim, dos 32 nós presentes no *dataset* externo (DtEsportes), após remover as duplicações, foram selecionados apenas um conjunto de 4 nós, que representa apenas 12% do *dataset* externo.

A nova mineração sobre o DtIME+ encontrou um total de 400 regras de associação de multirrelação em menos de 1 segundo. A Tabela 2 mostra algumas das novas regras que

foram geradas com as arestas identificadas e os esportes associados. É importante destacar que a regra "Supervised_By (Maria Cláudia), Study_In (IME) → Plays (Futebol)" não seria encontrada sem o processo de extensão do *dataset* DtIME. Essa regra, traduzindo para o português, diz que quem é supervisionado pela professora Maria Cláudia e estuda no IME, joga futebol, com o suporte de 10% e confiança de 77%. É possível observar, na Tabela 2, outras regras similares a essa relacionando outros professores. Isso pode sugerir que a instituição oferece alguma facilidade para prática desse esporte em suas dependências.

Tabela 2. Novas regras geradas, após a mineração do *dataset* DtIME+, com o algoritmo *MRAR+*.

Antecedente	Consequente	Conf.	Sup.
Plays(Vôlei de Praia)	Supervised_By (Work_On (IME))	1.00	0.28
Plays(Natação)	Live_In(RJ)	1.00	0.23
Study_In(IME), Plays(Natação)	Supervised_By (Work_On(IME))	0.87	0.20
Supervised_By(Maria Cláudia), Study_In(IME)	Plays (Futebol)	0.77	0.10
Supervised_By(Ronaldo Goldschmidt), Study_In(IME)	Plays (Futebol)	0.77	0.10
Supervised_By(Cláudia Justel), Study_In(IME)	Plays (Futebol)	0.77	0.10

Esse tipo de análise em um *dataset* maior com várias instituições e alunos envolvidos, poderia levar à descoberta das instituições que possuem facilidade em apoiar a prática de esportes. Além disso, em alguns casos, é possível identificar aquelas que apoiam esportes menos populares, como por exemplo, esgrima, tênis, etc.

Comparando os resultados obtidos com o *dataset* original, é possível ver que a análise realizada sobre o DtIME+ permitiu gerar o total de 400 regras, isto é 103 a mais que a análise anterior. Isso representa um aumento de aproximadamente 35% na quantidade regras geradas. Deste conjunto, 112 são regras novas; e 288 já tinham sido encontradas anteriormente.

7.2. Experimento 2 (Dados Sintéticos)

Este segundo experimento foi desenvolvido com o objetivo de validar o algoritmo *MRAR+* analisando um *dataset* de maior porte. Para isso foi necessário construir um novo conjunto de dados seguindo a mesma estrutura utilizada no *dataset* do experimento anterior, e utilizando uma ferramenta para popular o *dataset* com dados fictícios. Dentre as opções de ferramentas encontradas para geração de *datasets* com dados sintéticos, usamos o *Mockaroo Realistic Data Generator*[7], que já vem sendo utilizado pela comunidade

[7]https://mockaroo.com/

científica [Bytyçi et al. 2016, Acibar et al. 2016, Natali and Alfadian 2019]. No *Mockaroo*, os dados podem ser gerados respeitando-se um formato pré-definido, com base nas configurações informadas pelos usuários.

A versão livre do *Mockaroo* permite gerar até 1000 linhas por vez, com dados realistas. Além disso, eles podem ser gerados em diversos formatos, como por exemplo: CSV, JSON, SQL, Excel e muitos outros. A partir de 1000 linhas é necessário usar a versão paga[8]. Contudo, nada impede que o mesmo processo seja executado mais de uma vez, caso seja necessário um número maior de dados, ainda com a versão livre.

Para construir o *dataset* principal desse experimento, utilizamos novamente como modelo o grafo apresentado na Figura 1. Como já visto na descrição do experimento anterior, para construir um novo *dataset* é necessário que ele seja um grafo direcionado com rótulos nas arestas. Dessa maneira, iniciamos a construção do novo *dataset* respeitando o encadeamento entre os nós do grafo. Para cada aresta formada no grafo, o nó de destino foi usado novamente como nó de origem na aresta seguinte, compondo assim, uma cadeia de arestas e formando caminhos, conforme as definições 1 e 2 da Seção 4. Para fazer isso, foi necessário criar alguns conjuntos de dados, gerados aleatoriamente pelo *Mockaroo*, e em um segundo momento tais dados foram usados na criação dos relacionamentos entre eles.

Detalhando a construção do *dataset*, inicialmente foram criados conjuntos de dados no formato CSV de modo independente, i.e., sem estar correlacionados: alunos, universidades, professores, cidades, tipos de clima, projetos e tipos de condição de saúde. Também foi necessário criar uma lista com 3 colunas (alunos, professores e universidades) para garantir que o mesmo aluno só poderia estudar com um professor que tivesse vínculo com a mesma universidade.

Após gerar esses conjuntos de dados, que correspondem aos nós do grafo, foram configurados *schemas* no *Mockaroo* para relacionar um nó a outro, formando as arestas do grafo direcionado, ou seja, as triplas sujeito-predicado-objeto. Para isso, no menu *schemas* do *Mockarroo* foram especificados os tipos de dados selecionando os *datasets* gerados no passo anterior, e foi definida a relação a ser estabelecida entre eles. A Figura 8 ilustra como foi feito para gerar a relação "*Live_In*" entre os alunos e as cidades onde residem. Foram configurados três campos para permitir a geração das triplas: sujeito, predicado e objeto. O primeiro campo foi rotulado como "*Subject*", e seu tipo de dados como um "*Dataset Columm*", indicando a seguir o *dataset* "Aluno" e o campo "aluno" neste dataset, como fonte de dados. A opção de seleção *sequential* indica que a sequência de instâncias de alunos do *dataset* indicado será respeitada para a formação das triplas. O segundo campo foi rotulado como "*Predicate*", e o tipo de dados correspondente "*Template*" indica que é uma parte fixa a ser gerada, e que terá como rótulo a expressão "*Live_In*". Já o último campo foi rotulado como "*Object*", e o tipo de dados corresponde a um "*Dataset Columm*", indicando a seguir o *dataset* "Cidade" a ser usado como fonte de dados para a formação das triplas. A opção *random* é escolhida nesse caso para permitir valores em ordem aleatória. Após essa configuração bastou definir o número de instâncias a gerar ("*rows*") e escolher o formato de dados para exportação.

[8]https://www.mockaroo.com/profile/new

O mesmo processo foi feito para gerar as associações do tipo *Climate_Type*, relacionando cidades aos tipos de clima; *Patronage*, relacionando projetos e as universidades que o patrocinam; *Near_By*, que diz qual cidade fica próxima de outra; *Cooperator*, que diz que professor já cooperou em projetos de outro professor; *Supervised_By*, que informa quem supervisiona determinado aluno; *Work_On*, que informa qual professor trabalha em qual projeto; *Study_In*, que informa a universidade onde cada aluno estuda; e por fim, *Health_Condition*, que diz a condição de saúde de cada aluno. Sendo assim, ao juntar todos esses dados em um aquivo único um novo *dataset* foi criado e chamando de *DtMockaroo*, seguindo a mesma lógica do que foi visto na Figura 1.

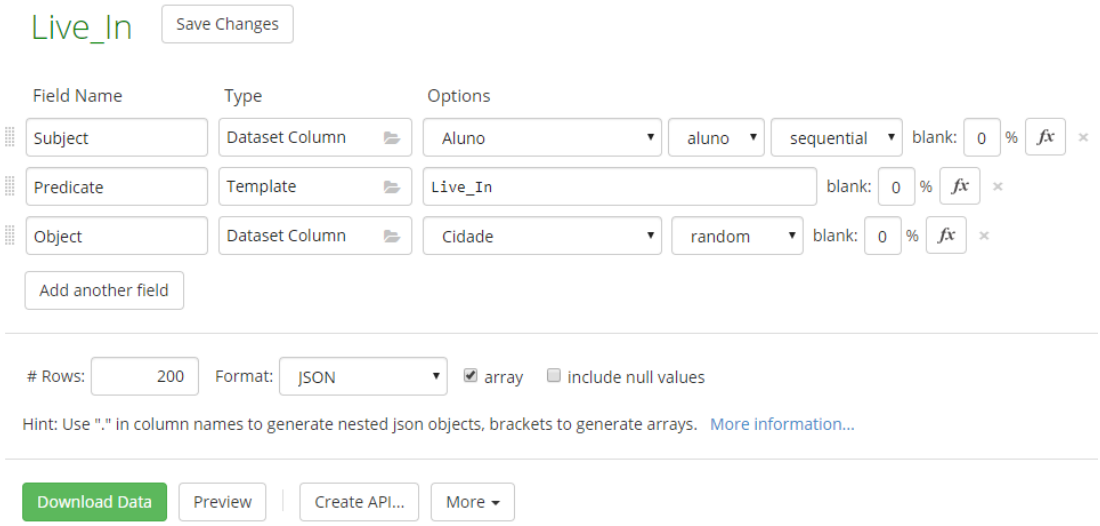

Figura 8. Tela do *Mockaroo* com a configuração do processo de geração da relação "*Live_In*".

Como o objetivo desse experimento é demostrar a utilização completa do *MRAR+*, o que inclui o acesso a um *dataset* real na Web de Dados, foi necessário associar os recursos compostos pelos nós que representam cada um dos alunos à URI que representa uma pessoa real no *dataset* do *DBpedia*, através da relação *sameAs*. Para isso, foi necessário realizar a seguinte consulta em SPARQL "SELECT distinct ?s WHERE { ?s a foaf:Person. ?s dct:subject dbr:Category:Brazilian_footballers. ?s dbo:team dbr:Santos_FC }", com o objetivo de identificar uma lista com jogadores que tivessem algum clube em comum.

Sendo assim, para este experimento, foi então utilizado o *dataset DtMockaroo*, composto por 449 nós e 1000 arestas, aproximadamente 12,5 vezes maior do que o *dataset* do primeiro experimento. Inicialmente, no passo 1, o algoritmo MRAR é executado sobre o *DtMockaroo*, com a seguinte configuração de parâmetros: *MinSup* = 5%, *MinConf* = 60%, *MinLevel* = 1 e *MaxLevel* = 4. Como resultado foram encontradas 15 regras. Dentre elas, vale destacar a regra Supervised_By (Work_On (Patronage (UNIFESP))), Supervised_By (Work_On (Project_A)) → Supervised_By (Work_On (Patronage (USP))). Essa regra, traduzindo para o português, diz que quem é supervisionado por alguém que trabalha em um projeto patrocinado pela UNIFESP e é supervisionado por alguém que trabalha no Projeto_A, é supervisionado por alguém que trabalha em um projeto patrocinado pela

USP, com suporte de 8% e confiança de 100%. Isso pode sugerir que a USP e a UNIFESP patrocinam um projeto em comum e que seus professores são colaboradores.

É importante destacar que para ter acesso ao *dataset* externo, foi utilizado o *endpoint* de consulta do *DBpedia*[9]. Por se tratar de um *endpoint* de grande porte, com milhões de triplas, consultas genéricas em *SPARQL*, sem muitas restrições, podem trazer um volume muito grande de resultados. Por exemplo, uma consulta do tipo *SELECT ?p ?o WHERE { ?suporte ?p ?o }*, onde o termo "?suporte" é, na verdade, uma variável que recebe os recursos externos que dão suporte às regras geradas no passo 1, causaria uma ampliação muito grande do *dataset* alvo, inviabilizando a análise posterior (passo 4). Portanto, neste experimento, visando controlar o volume de informações vindas do *dataset* externo, no passo 3, foi acrescentada à consulta genérica em *SPARQL* uma restrição para filtrar os predicados/relações. O filtro utilizado para este experimento selecionou apenas os clubes dos quais os jogadores de futebol (recursos de suporte) já haviam feito parte, por meio de uma consulta similar a "*SELECT ?team WHERE {?jogador dbo:team ?team }*".

A seguir, ao acionar o botão *MRAR+*, com base nas regras encontradas no passo 1 e seus recursos (nós) de suporte, o algoritmo *MRAR+* identifica os recursos externos (passo 2), o que viabiliza o processo de extensão dos dados (passo 3). Neste passo, o *dataset* alvo (DtMockaroo) é estendido gerando um novo *dataset*, o *DtMockaroo+*, contendo os dados originais somados às propriedades e objetos que foram encontradas no *dataset* externo, o *DBpedia*. Após essa ampliação controlada, o *dataset* estendido passa a possuir um total de 925 nós e 2302 arestas, o que representa um aumento de 206% em número de nós e 230% em relação ao número de arestas, comparado ao *dataset* alvo original. É importante destacar que esse processo permitiu identificar os recursos externos com potencial de gerar novas regras, uma vez que os nós selecionados estavam diretamente ligados aos recursos que deram suporte às primeiras regras encontradas.

Sendo assim, dando continuidade ao experimento, fez-se nova análise sobre o *dataset* ampliado, a fim de encontrar novas regras de associação de multirrelação que pudessem associar os recursos locais com os externos. Tendo em vista que o volume de dados obtidos superou em quantidade o volume do *dataset* alvo, foi necessário fazer uso do recurso oferecido pelo algoritmo para calcular um suporte que permita que novas regras sejam geradas, i.e., a opção "*Apply the best support*", cuja formula é descrita na Seção 6. Dessa maneira uma nova mineração foi realizada mantendo as mesmas configurações anteriores, só ajustando o novo *MinSup* para 2,4%.

A nova análise sobre o *DtMockaroo+* encontrou um total de 55 regras de associação de multirrelação. A Tabela 3 mostra algumas das novas regras que foram geradas com as arestas identificando os clubes de futebol onde os recursos passaram a estar associados. É importante destacar que a regra "Supervised_By (Work_On (Project_E)), http://dbpedia.org/ontology/team (http://dbpedia.org/resource/Santos_FC) $\rightarrow$ Supervised_By (Work_On (Patronage (PUC-SP)))" não seria encontrada sem o processo de extensão do *dataset* alvo. Essa regra, traduzindo para o português, diz que quem é supervisionado por alguém que trabalha no Projeto_E e joga no Santos_FC, é supervisionado

[9]http://dbpedia.org/sparql/

por alguém que trabalha em um projeto patrocinado pela PUC-SP, com suporte de 5% e confiança de 100%. Isso pode sugerir que a PUC-SP pode patrocinar um projeto do clube Santos_FC.

Tabela 3. Novas regras geradas, após a mineração do *dataset DtMockaroo+*, com o algoritmo *MRAR+*. Para melhorar a visualização a URI "http://dbpedia.org/ontology/" foi abreviado para "dbo:" e "http://dbpedia.org/resource/" para "dbr:". Vale ressaltar que todas as regras desta tabela são meramente ilustrativas, geradas a partir de dados sintéticos.

Antecedente	Consequente	Conf.	Sup.
Supervised_By (Work_On (Project_E)), dbo:team (dbr:Santos_FC)	Supervised_By (Work_On (Patronage (PUC-SP)))	1.00	0.05
Live_In (Climate_Type (Hot)), dbo:team (dbr:Santos_FC)	Live_In (São Paulo)	1.00	0.04
Supervised_By (Work_On (Patronage (UNIFESP))), Health_Condition (Good)	dbo:team (dbr:Santos_FC)	1.00	0.04
dbo:team (dbr:Brazil_national_football_team)	dbo:team (dbr:Santos_FC)	1.00	0.03

Comparando os últimos resultados obtidos com o *dataset* original, é possível ver que a análise realizada sobre o *DtMockaroo+* permitiu gerar o total de 55 regras, sendo 40 delas regras novas, que não seriam encontradas apenas com a primeira análise. Isso representou um aumento de aproximadamente 267% na quantidade regras novas. Além disso, através desses experimentos foi possível perceber que as regras geradas podem ser úteis na descoberta de correlação entre recursos dos distintos *datasets*, evidenciando o potencial dessa abordagem para o enriquecimento da Web de Dados com novas ligações semânticas.

8. Conclusões e perspectivas futuras

A integração consistente de dados na Web é capaz de gerar conhecimentos novos e úteis. No entanto, o crescente número de *datasets* disponíveis na Web torna essa integração um dos principais desafios da Web de Dados. Esse artigo apresenta uma solução para viabilizar a busca de informações a partir de *datasets* interligados, com vistas a facilitar uma integração maior entre esses datasets. Isso é feito através do uso de mineração de regras de associação de multirrelação em grafos. O método proposto analisa um *dataset* alvo escolhido e, a partir desse, outros *datasets* com potenciais para enriquecer os resultados da mineração do *dataset* alvo.

Outras contribuições desse trabalho foram a formalização da técnica de mineração de regras de associação de multirrelação (apresentado na Seção 4) e a implementação do algoritmo *MRAR+* (desenvolvido como uma extensão do algoritmo MRAR), apresentado na Seção 6.

Os experimentos apresentados nas Subseções 7.1 e 7.2 mostraram que as regras novas obtidas através do MRAR+ podem ampliar o conhecimento sobre os dados analisados nos experimentos. Vale ressaltar, que o segundo experimento comprova a funcionalidade completa do MRAR+ e a sua escalabilidade. Nesse experimento utilizou-se um *dataset* de maior porte, gerado sinteticamente, e fez-se acesso a tuplas de um *endpoint* da Web de Dados para a ampliação desse dataset. Adicionalmente, a partir das regras geradas pelos experimentos, foi possível vislumbrar novas ligações semânticas entre os datasets, evidenciando o potencial do MRAR+ em contribuir para o enriquecimento das ligações na Web de Dados.

A implementação da abordagem *MRAR+* ainda é um protótipo em evolução, ganhando novas versões na medida em que novas funcionalidades vão sendo incorporadas. Uma das limitações encontradas na abordagem *MRAR+* é que, para alguns datasets, é preciso usar um suporte muito baixo para obter regras úteis. Essa limitação já está sendo tratada. Além disso, trabalhos futuros incluem a realização de outros experimentos com *datasets* reais, e a exploração de extensões do *dataset* alvo, trazendo caminhos de maior comprimento provenientes do *dataset* externo. Outro trabalho futuro importante será um estudo comparativo entre os resultados gerados pela *MRAR+* e os produzidos pela mineração de dados aplicada a um *dataset* resultante do processo de integração completa de *datasets* interligados. Espera-se que tal estudo permita avaliar de forma quantitativa a otimização proporcionada pela *MRAR+* em relação ao tempo de processamento e de forma qualitativa a cobertura das regras mineradas. Por fim, pretende-se incorporar as funcionalidades dos algoritmos MRAR e *MRAR+* a sistemas gerenciadores de banco de dados em grafo (SGBDG), com o objetivo de permitir que esse tipo de análise possa ser feita de forma rápida como recurso interno do próprio SGBDG.

Referências

Acibar, J., Aguanta, L., Gomora, J., and Velasco, L. (2016). Data analysis with visualization for a geographic information system of schistosomiasis community health data. In *Pre-proceeding of the 6th Workshop on Computation: Theory and Practice WCTP*.

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216.

Ait-Mlouk, A., Jiang, L., and Vu, X.-S. (2019). Improving rdf data through semantic association rules mining. In *31st Sweedish AI Society Workshop (SAIS 2019)*.

Barati, M. (2019). *Automated Knowledge Enrichment for Semantic Web Data*. PhD thesis, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand.

Barati, M., Bai, Q., and Liu, Q. (2017). Mining semantic association rules from RDF data. *Knowl.-Based Syst.*, 133:183–196.

Basse, A., Gandon, F., Mirbel, I., Lo, M., Mirbel, I., and Fr (2010). Dfs-based frequent graph pattern extraction to characterize the content of rdf triple stores.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, 284(5):34–43.

Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *Int. journal on Semantic Web and Information Systems*, 5(3):1–22.

Bizer, C., Heath, T., Idehen, K., and Berners-Lee, T. (2008). Linked data on the web (ldow2008). In *Proceedings of the 17th Int. Conf. on World Wide Web*, WWW '08, pages 1265–1266, New York, NY, USA. ACM.

Bytyçi, E., Ahmedi, L., and Kurti, A. (2016). Association rule mining with context ontologies: An application to mobile sensing of water quality. In Garoufallou, E., Subirats Coll, I., Stellato, A., and Greenberg, J., editors, *Metadata and Semantics Research*, pages 67–78, Cham. Springer International Publishing.

de Oliveira, F. A., Costa, R. L., Goldschmidt, R. R., and Cavalcanti, M. C. (2019). 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*, pages 61:1–61:8.

Elseidy, M., Abdelhamid, E., Skiadopoulos, S., and Kalnis, P. (2014). Grami: Frequent subgraph and pattern mining in a single large graph. volume 7, pages 517–528.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37.

Goldschmidt, R., Bezerra, E., and Passos, E. (2015). *Data Mining: Conceitos, técnicas, algoritmos, orientações e aplicações*. Rio de Janeiro: Elsevier.

Hendrickx, T., Cule, B., Meysman, P., Naulaerts, S., Laukens, K., Goethals, B.", e. T., Lim, E., Zhou, Z., Ho, T., Cheung, D., and Motoda, H. (2015). *Mining Association Rules in Graphs Based on Frequent Cohesive Itemsets*, pages 637–648. Springer Int. Publishing, Cham.

Natali, V. and Alfadian, P. (2019). Analisis dan perancangan domain specific language untuk data generator pada relational database. *JUMANJI (Jurnal Masyarakat Informatika Unjani)*, 3(01):64–73.

Oliveira, F. A. (2018). Mineração de regras de associação de multirrelação em datasets na web de dados. Master's thesis, Instituto Militar de Engenharia(IME), Rio de Janeiro.

Oliveira, F. A., Costa, R. L., Goldschmidt, R. R., and Cavalcanti, M. C. (2017). Mineração de regras de associação multirrelação em grafos: Direcionando o processo de busca. In *Simpósio Brasileiro de Banco de Dados (SBBD'17)*, pages 270–275.

Pickler, M. E. V. (2007). Web semântica: ontologias como ferramentas de representação do conhecimento. *Perspectivas em Ciência da Inf.*, 12(1):65–83.

Raad, J., Beek, W., van Harmelen, F., Pernelle, N., and Saïs, F. (2018). Detecting erroneous identity links on the web using network metrics. In Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M. C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.-A., and Simperl, E., editors, *The Semantic Web – ISWC 2018*, pages 391–407, Cham. Springer International Publishing.

Ramezani, R., Saraee, M., and Nematbakhsh, M. A. (2014). MRAR : Mining Multi-Relation Association Rules. *J. of Computing and Security*, 1(2):133–158.

Rehman, S. U., Khan, A. U., and Fong, S. (2012). Graph mining: A survey of graph mining techniques. In *Seventh International Conference on Digital Information Management (ICDIM 2012)*, pages 88–92.

Sydow, M., Pikuła, M., and Schenkel, R. (2013). The notion of diversity in graphical entity summarisation on semantic knowledge graphs. *J Intell Inf Syst*, 41:109–149.

Tavares, A., Oliveira, H., and Lóscio, B. (2015). Rdfmat–um serviço para criação de repositórios de dados rdf a partir de crawling na web de dados. *Revista da Escola Regional de Informática*, 1(1):6.

Vieira, M. R., FIGUEIREDO, J. M. d., Liberatti, G., and Viebrantz, A. F. M. (2012). Bancos de dados nosql: conceitos, ferramentas, linguagens e estudos de casos no contexto de big data. *Simpósio Brasileiro de Bancos de Dados*, 27:1–30. 27/12/2017.

Zhang, X., Zhao, C., Wang, P., and Zhou, F. (2012). Mining link patterns in linked data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7418 LNCS:83–94. cited By 10.