Método para Rotular Ligações Semânticas na Web de Dados

Rafael Neves da Silveira[1], **Maria Cláudia Cavalcanti**[1]

[1] Departamento de Sistemas e Computação
Instituto Militar de Engenharia (IME) – Rio de Janeiro, RJ – Brasil

{rafa.ns,yoko}@ime.eb.br

Abstract. *The Semantic Web, with its languages and standards, provides a common framework that allows data to be shared and reused. One way to increase knowledge about this data is by making new interconnection between datasets. However, most of the interconnection approaches have connections like "Same As"or "Related To". The latter type leaves vague the meaning of the relationship found. This paper presents a method to label this type of relation between datasets, through the use of ontologies and controlled vocabularies. Besides the method, it also presents the WEB application called PLAIN that implements it, and a case study demonstrating the feasibility and functionality of the proposed approach.*

Resumo. *A Web Semântica, com suas linguagens e padrões, fornece uma estrutura comum que permite que os dados sejam compartilhados e reutilizados. Uma forma de aumentar o conhecimento sobre esses dados é realizando novas interligações entre datasets. No entanto, a maioria das abordagens de interligação apresentam ligações do tipo "Same As" ou "Related To". Este último tipo deixa vago o significado da relação encontrada. Este trabalho apresenta um método para rotular esse tipo de relação entre datasets, por meio da utilização de ontologias e vocabulários controlados. Além do método, apresenta também a aplicação WEB denominada PLAIN que o implementa, e um estudo de caso demonstrando a viabilidade e funcionalidade da abordagem proposta.*

1. Introdução

A Web Semântica[1] (WS), com suas linguagens e padrões, fornece uma estrutura comum que permite que os dados sejam compartilhados e reutilizados além dos limites de aplicativos, empresas e comunidades. A iniciativa denominada Dados Conectados, do inglês *Linked Data*, é um conjunto de boas práticas para a publicação de dados na Web. Nesse contexto, os Dados conectados são dados publicados e ligados utilizando as tecnologias e padrões da WS [Yu 2014]. Uma forma de aumentar o conhecimento sobre esses dados é realizando uma ampliação dos *datasets* da WS. No entanto, essa não é uma tarefa simples. A partir de um *dataset* de interesse, que pode ser chamado de *dataset* Fonte, a tarefa de ampliação inicia-se pela busca por *datasets* externos que contenham recursos comuns ao *dataset* Fonte.

Para realização dessa tarefa, algoritmos baseados em mineração de regras de associação são eventualmente utilizados em *Data Mining* e, em geral, acarretam em uma grande quantidade de regras geradas. Por exemplo, o algoritmo MRAR+

[1] https://www.w3.org/standards/semanticweb/

[de Oliveira et al. 2019], uma evolução do algoritmo MRAR [Ramezani et al. 2014], explora regras de associação de multirrelação sobre grafos direcionados, como os que são usados para representação de dados na Web de Dados.

Atualmente, as relações encontradas apontam para pares de recursos que podem ter alguma relação, sendo que cada recurso de um par pertence aos *datasets* Fonte e Alvo. A maioria das ferramentas de descoberta de relações são voltadas para as que são do tipo *"Same As"*. No entanto, a semântica dessas relações não é identificada, e consequentemente, não é possível rotular tais relações de uma forma semanticamente clara. Como em [Sherif et al. 2015], vê-se o uso de relações do tipo *"Related To"*.

Sendo assim, este trabalho visa apresentar um método para enriquecer as ligações entre o *dataset* fonte e os *datasets* externos com os quais ele se relaciona. Isso se dá por meio da explicitação da semântica dessas ligações e da rotulação das mesmas. A ideia é partir das regras produzidas por algoritmos de mineração na Web de Dados, e utilizando catálogos de vocabulários e ontologias, encontrar recursos semânticos que ajudem a rotular as ligações. A proposta inclui também o uso de técnicas de NLP, mais especificamente, técnicas de Extração de Relações (RE). Uma implementação é apresentada, e um estudo de caso ilustra a viabilidade do método proposto, mostrando resultados promissores.

O trabalho está organizado como se segue. A próxima seção apresenta os conceitos relevantes para compreensão da proposta deste artigo. A Seção 3 apresenta os trabalhos relacionados. Já a Seção 4 explicita o método proposto e a Seção 5 a sua implementação. A Seção 6 apresenta um estudo de caso utilizado nessa abordagem, e a Seção 7 destaca as contribuições e aponta para trabalhos futuros.

2. Conceitos Básicos

A Web Semântica (WS) estende a Web tradicional [Berners-Lee et al. 2001]. Vocabulários controlados e ontologias fazem parte da proposta da WS, adicionando significado ao conteúdo da Web tradicional. Um vocabulário é um conjunto de termos inequivocamente definidos, utilizados na comunicação [Hebeler et al. 2011]. Já uma ontologia é definida como uma especificação formal e explícita de uma conceituação compartilhada [Studer et al. 1998], que utiliza um vocabulário pré-definido e reservado de termos para definir conceitos e as relações entre eles, dentro de dado um domínio [Hebeler et al. 2011].

As linguagens de representação de ontologias, RDF[2] e OWL[3] (*Ontology Web Language*), tornaram-se padrão para a materialização da WS. Essas linguagens têm sido consideradas tecnologias-chave na WS, pois facilitam a interpretação da informação disponível na Web tradicional, por agentes de computação. Elas também são usadas para representar dados estruturados na Web (Web de Dados). Segundo apresentado em [Horrocks 2008], o RDF é uma linguagem cuja estrutura de dados básica é um grafo rotulado direcionado, e sua única construção sintática é a tripla, que consiste de três componentes, chamados de sujeito, predicado e objeto. Uma tripla representa uma aresta (denominada predicado) conectando dois nós (denominados sujeito e objeto), ou seja, descreve uma relação entre sujeito e objeto através do predicado. Um dos predicados mais importantes é o *rdf:type*, que representa a relação classe-instância.

[2]https://www.w3.org/RDF/
[3]https://www.w3.org/OWL/

A linguagem RDF inclui o RDFS (*RDF Schema*) que fornece um vocabulário específico usado para definir classes, propriedades e especificações simples de domínio (*rdfs:domain*) e de alcance (*rdfs:range*) dessas propriedades. Já a linguagem OWL, conforme [Hebeler et al. 2011], estende o vocabulário RDFS com recursos adicionais que podem ser usados para construir ontologias mais expressivas para a Web. A OWL introduz restrições adicionais em relação à estrutura e ao conteúdo dos documentos em RDF a fim de tornar o processamento e o raciocínio (inferência) mais conclusivos computacionalmente. A seguir é apresentado um exemplo de triplas RDF e que utiliza também um predicado da linguagem OWL. A primeira tripla representa a relação de equivalência entre duas classes, *Person* e *Human*. Já a segunda tripla declara que *Mary* é uma instância da classe *Person*. Por fim, a terceira tripla ilustra a capacidade de inferência da linguagem OWL, que a partir da informação declarada das duas primeiras triplas, é possível inferir que *Mary* é uma instância da classe *Human*.

Person	*owl:equivalentClass*	*:Human*	.
Mary	*rdf:type*	*:Person*	.
Mary	*rdf:type*	*:Human*	. *(inferência)*

Para realização de consultas em *datasets* cujos dados estão armazenados ou são visualizados em RDF, é utilizada uma linguagem chamada de SPARQL[4]. Os *datasets* em RDF são geralmente disponibilizados através de interfaces de consulta SPARQL, acessíveis por seus endereços Web, conhecidos como *Endpoints*. Segundo apresentado em [Laufer 2015], para que o cenário da WS ficasse completo foi preciso estabelecer um conjunto de vocabulários de referência de forma a facilitar o reuso, e consequentemente, o alinhamento entre os metadados. Existem alguns catálogos que podem auxiliar o usuário na busca por vocabulários ou ontologias a reusar, entre eles destaca-se o LOV[5] (*Linked Open Vocabularies*). O catálogo LOV está armazenado em um *dataset* disponível para consultas através de um *endpoint SPARQL*. Atualmente, O LOV contém mais de 700 vocabulários e está em constante crescimento [Vandenbussche et al. 2017].

O uso de vocabulários e ontologias de referência facilita a interligação de *datasets*. Uma iniciativa chamada Nuvem de Dados Abertos e Conectados[6], LOD (*Linked Open Data*), também tem contribuído no sentido de aumentar a interligação entre *datasets* na Web de Dados [Assaf et al. 2015]. Até maio de 2020, tinha-se 1.255 conjuntos de dados com 16.174 *links*. Estudos mostram que 44% dos *datasets* em LOD não estão conectados a outros conjuntos de dados [Schmachtenberg et al. 2014]. Conforme apontado em [Nentwig et al. 2017], o principal motivo dessa importante falta de links na nuvem LOD está na dificuldade de criá-los, sendo um processo muito custoso quando realizado manualmente. Segundo os autores, a maioria das abordagens tratam o problema da descoberta de links como um problema de computação de similaridade. Dados dois conjuntos de recursos, nós do grafo, Fonte (do inglês *Source* – S) e Alvo (do inglês *Target* – T), ambos são interligados por propriedades dentro de cada *dataset*. Em [Schmachtenberg et al. 2014] os autores explicam que a descoberta de link, do inglês *Link Discovery* (LD), tem como objetivo encontrar automaticamente pares de recursos em $S \times T$ que devem ser vinculados entre si, por exemplo, com um relacionamento do tipo *owl:sameAs*.

[4]https://www.w3.org/TR/sparql11-overview/
[5]https://lov.linkeddata.es/dataset/lov/
[6]https://lod-cloud.net/

Conforme [Nentwig et al. 2017], esse problema pode ser descrito como segue. Dados dois conjuntos de recursos (*datasets*) S e T e uma relação R (por exemplo, *owl:sameAs*), encontre todos os pares $(s, t) \in S \times T$ tais que $R(s, t)$ se mantenha. O resultado é representado como um conjunto de links chamado mapeamento: $M_{S,T} = \{(s_i, R, t_j)|s_i \in S, t_j \in T\}$;

Datasets como DBpedia[7] ou LinkedGeoData[8] geralmente utilizam uma ontologia que descreve seus recursos e os interligam através de propriedades pré-definidas. Conforme pode ser observado na LOD, o fato desses *datasets* alvos utilizarem ontologias faz com que os *datasets* fontes passem também a utilizá-las, facilitando a interligação. Nesse sentido, em [Paris 2018] o autor explica que uma das propriedades mais importantes da LOD é a *owl:sameAs*, que é usada para indicar que dois recursos são iguais. Em [de Oliveira et al. 2019], quando esses recursos provêm de dois *datasets* diferentes, a interligação é chamada de externa e possibilita a realização de consultas que alcançam ambos os *datasets*. Nesse contexto, em [Athanasiou et al. 2019] os autores definem o processo de enriquecimento de *dataset* como um conjunto de ações para identificar e recuperar informações adicionais relacionadas aos recursos do *dataset* de origem a partir de fontes externas. Esse processo cria propriedades extras relacionadas a esses recursos, aumentando a riqueza e completude dos dados.

3. Trabalhos Relacionados

Muitos trabalhos têm realizado enriquecimento de *datasets* da Web de dados através de interligações entre eles. O Silk [Bizer et al. 2009] é uma ferramenta que se apoia em regras especificadas manualmente e aprendizado supervisionado para a produção de links *owl:sameAs* ou outros relacionamentos especificados pelo usuário. Ela utiliza métricas de similaridade de *strings* para realização da tarefa de interligação.

Já o LIMES [Ngomo and Auer 2011] é uma ferramenta que suporta tanto a configuração manual quanto as técnicas de aprendizado supervisionadas e não supervisionadas. A ferramenta oferece diferentes técnicas de aproximação baseadas em espaços métricos para estimar as semelhanças entre instâncias. Assim como o Silk, ele pode produzir links *owl:sameAs* ou especificados pelo usuário. Ainda no contexto de uso do LIMES, em [Sherif et al. 2015], os autores apresentaram o componente de enriquecimento chamado de DEER. O algoritmo apresentado utiliza aprendizado de máquina supervisionado para realização do enriquecimento dos dados utilizando metadados que estão implícitos no *dataset* fonte do enriquecimento, porém não consulta dados externos à ele para realização da tarefa. Já em [Ahmed et al. 2019], os autores apresentam uma abordagem baseada em sumarização para descrever os links encontrados.

Nesse contexto, em [de Oliveira et al. 2019], os autores propõem um algoritmo para minerar regras de associação de multirrelação em mais de um *dataset*, chamado de MRAR+, desenvolvido como uma extensão do MRAR [Ramezani et al. 2014]. No trabalho, a descoberta de regras de associação na Web de Dados foi viabilizada por meio da atribuição de uma máscara de busca durante o processo de mineração. Isso fez com que o algoritmo gerasse apenas as regras que estavam relacionadas aos recursos de maior frequência e que estivessem vinculados a recursos de *datasets* externos, reduzindo assim o

[7]https://wiki.dbpedia.org/
[8]http://linkedgeodata.org/

custo computacional. O resultado desse algoritmo é um conjunto de regras que associam recursos que possuem potencial para interligação. A observação dessa oportunidade de enriquecimento serviu de motivação para o desenvolvimento do presente trabalho.

Até onde foi possível investigar, nenhum dos trabalhos relacionados acima identifica claramente a semântica das novas relações descobertas (exceto relações do tipo *owl:sameAs*), e consequentemente, não é possível rotular tais relações com clareza. Sendo assim, o presente trabalho visa preencher essa lacuna, como descrito na próxima seção.

Uma outra linha de trabalhos relacionados vem contribuindo para a construção da WS através o uso de Processamento de Linguagem Natural, do inglês *Natural Language Processing* – NLP. Por exemplo, através de técnicas de NLP é possível identificar nos textos das páginas Web referências a entidades do mundo real e suas relações, e atribuir uma *Uniform Resource Identifier* – URI a cada uma dessas entidades. Essa técnica específica é conhecida como *Information extraction* (IE). Além de identificar entidades do mundo real, chamadas Entidades Nomeadas (ou *Named Entities* – NEs, em inglês), a IE deve ser capaz também de extrair as relações entre elas, tarefa conhecida como *Relation Extraction* – RE. As NEs são geralmente consideradas as principais componentes do texto, podendo ser pessoas, locais, organizações, nomes próprios ou expressões temporais.

Para apoiar a tarefa de RE e identificar links entre as NEs, em [Devlin et al. 2018] os autores explicam que o desenvolvimento de um modelo de linguagem pré-treinado mostra-se efetivo. Nesse trabalho, os autores apresentam o modelo BERT (*Bidirectional Encoder Representations from Transformers*), cujos resultados estão de acordo com recentes avanços do NLP. Em [Collovini et al. 2020] os autores apresentaram o RelP++, um framework que combina Reconhecimento de NEs e RE para o português. Já em [Han et al. 2019] apresenta-se o OpenNRE, um conjunto de ferramentas para desenvolver e aplicar modelos de RE.

Como foi visto, a técnica de RE é normalmente aplicada a conjuntos de textos. No entanto, o presente trabalho vislumbra a sua aplicabilidade no enriquecimento semântico das ligações entre *datasets* na Web de Dados. Assim, o método descrito a seguir procura aliar a técnica de RE com este objetivo, indo além da informação contida nos *datasets* e recursos semânticos, como ontologias e vocabulários.

4. O Método *Predicate Labeling*

O problema que buscamos tratar é caracterizado pela ausência de semântica observada nos links gerados durante o processo de interligação de *datasets*. Observando os trabalhos publicados é possível notar que, após a mineração de links, outras relações podem ser inferidas e devidamente rotuladas. Os processos do estado da arte focam no reconhecimento das relações mas deixam de realizá-las ou o fazem de forma limitada e sem o devido reconhecimento semântico. Haveria uma alternativa para realizar esse enriquecimento? Diante desse problema, a hipótese levantada é de que o uso de ontologias e vocabulários controlados combinado a técnicas de RE, podem favorecer a identificação e concepção dessas relações.

O método proposto, denominado Predicate Labeling, parte de um conjunto de relações que precisam de um rótulo com mais semântica e busca apoio nos vocabulários e ontologias existentes para sugerir novos rótulos. O método Predicate Labeling toma como base as seguintes definições:

Def.1 Sejam os seguintes conjuntos:
- $c_i \in C$, i.e., C é um conjunto formado por classes c_i.
- $e_i \in I$, onde I é um conjunto de instâncias e_i, tal que para cada e_i, pode existir um par $(e_i, c_j) \wedge c_j \in C$, i.e., e_i é instância da classe c_j;
- $x_i \in X$, onde X é um conjunto de recursos x_i, tal que cada recurso x_i pode ser uma classe ($x_i \in C$) ou uma instância ($x_i \in I$);
- $r_k \in R$, onde R é um conjunto de relações r_k que interligam recursos (sujeitos) a outros recursos (objetos).

Def.2 A tripla (x_i, r_k, x_j), onde a relação $r_k \in R$ liga o recurso x_i ao x_j, e $x_i, x_j \in X$.

Def.3 Sejam S e T dois *datasets* (Seção 2). O conjunto de recursos D é um *dataset* cujos elementos são triplas (x_i, r_k, x_j), tais que $(x_i, x_j) \in S \times T$ (Seção 2).

Def.4 A função $subClassOf()$ é uma função tal que, dadas as classes $c_i, c_j \in C$, se existir a tripla (c_i, *rdfs:subClassOf*, c_j), então $subClassOf(c_j) = c_i$.

Def.5 A função $superClassOf()$ é uma função tal que, dadas as classes $c_i, c_j \in C$, se existir a tripla (c_j, *rdfs:subClassOf*, c_i), então $superClassOf(c_i) = c_j$.

Def.6 A função $ancestorOf()$ é uma função tal que, dadas as classes $c_i, c_j \in C$, $ancestorOf(c_i)$ tem como resultado todas as classes superiores no ramo hierárquico em que se encontra c_i, ou mais formalmente, $ancestorOf(c_i) = \{c_j | \exists c_j = superClassOf(c_i)\} \cup ancestorOf(c_j)$.

Def.7 A função $descendantOf()$ é uma função tal que, dadas as classes $c_i, c_j \in C$, $descendantOf(c_i)$ tem como resultado todas as classes inferiores no ramo hierárquico em que se encontra c_i, ou mais formalmente, $ancestorOf(c_i) = \{c_j | \exists c_j = subClassOf(c_i)\} \cup descendantOf(c_j)$.

Def.8 A função $equivalentClass()$ é uma função tal que, dada uma classe $c_i \in C$, $equivalentClass(c_i)$ obtém como resultado todas as classes $c_j \neq c_i$ que contenham o mesmo conjunto de instâncias I.

Def.9 O conjunto C_{x_i} é um conjunto de classes que resulta da aplicação das funções definidas anteriormente (**Def.6**. **Def.7** e **Def.8**), mais formalmente pode-se dizer que $C_{x_i} = \{c_i\} \cup ancestorOf(c_i) \cup descendantOf(c_i) \cup equivalent(c_i)$.

Def.10 A função $domain^{-1}()$ é uma função tal que, dada uma classe $c_i \in C$, obtém-se como resultado todas as relações r_k que podem ter c_i como sujeito. Mais formalmente, $domain^{-1}(c_i) = \{r_k \in R \,|\exists (c_i, r_k, x_j), x_j \in X\}$.

Def.11 A função $range^{-1}()$ é uma função tal que, dado um recurso $x_j \in X$, obtém-se como resultado todas as relações r_k que podem ter x_j como objeto. Mais formalmente, $range^{-1}(x_j) = \{r_k \in R \,|\exists (x_i, r_k, x_j), x_i \in X\}$.

A Figura 1 apresenta uma visão geral do método *Predicate Labeling*, através de um diagrama que utiliza a notação BPMN[9] (*Business Process Model and Notation*). Inicia-se o processo com a atividade **Ler Base de Dados Conectados** que faz leitura do *dataset* D (**Def.3**), que contém os dados de origem para a rotulação. Nesta etapa são obtidas as triplas (**Def.2**) $(x_i, r_k, x_j) \in D$ cujas relações r_k precisam ser rotuladas, pois possuem semântica pobre.

O fluxo principal então segue para a atividade **Consultar Classes e Propriedades correlatas utilizando recursos semânticos** que é onde acontece a exploração de ontologias e catálogos de vocabulários controlados. Todo o fluxo de atividades dessa etapa está detalhado no diagrama da Figura 2.

[9]http://www.bpmn.org/

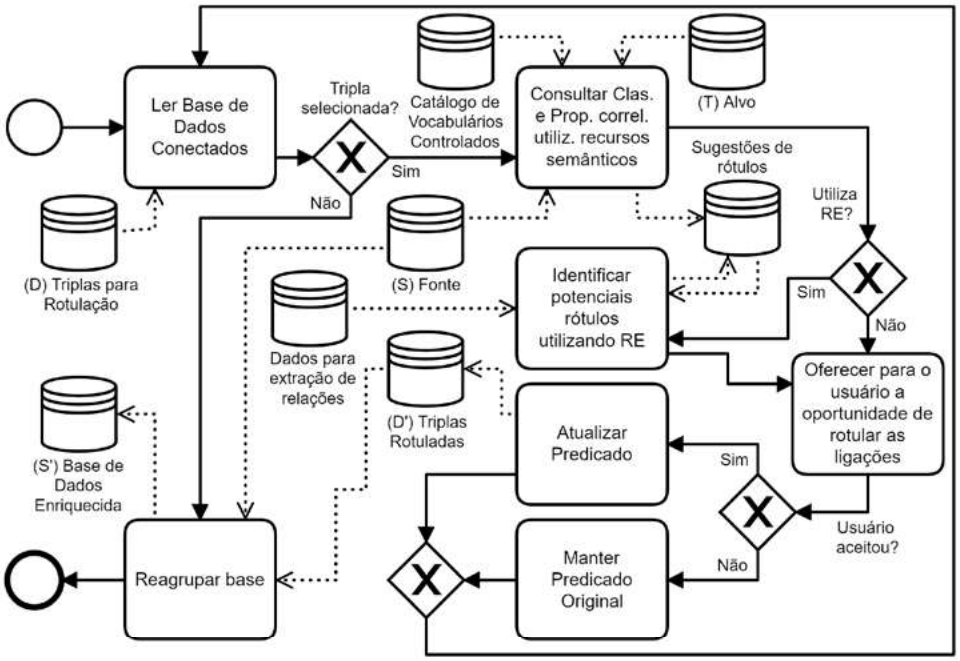

Figura 1. Diagrama do método Predicate Labeling em BPMN.

A sequência da atividade **Verificar se o recurso é Classe ou Instância** se desdobra em duas atividades paralelas exclusivas. Para cada recurso x_i em triplas $(x_i, r_k, x_j) \in D$, caso $x_i \in I$, o fluxo segue para a atividade **Buscar Classe mapeada na Base de Dados de Origem** que realiza uma busca por c_i que fora previamente mapeada em S. Caso esse mapeamento exista ou $x_i \in C$, ele será aproveitado nas consultas seguintes da atividade **Buscar Classes equivalentes**, com o uso das funções *ancestorOf()*, *descendantOf()* e *equivalentClass()* (**Def.6**, **Def.7** e **Def.8**, respectivamente). Por outro lado, se $x_i \in I$ e a informação sobre a sua classe não estiver formalizada em S, o fluxo paralelo exclusivo segue para a atividade **Buscar Classe em Catálogos de Vocabulários Controlados**, de forma a auxiliar nessa formalização. Com a aplicação dessas funções sobre c_i é construído o conjunto C_{x_i}, conforme definido em **Def.9**.

Analogamente à busca feita por c_i, que corresponde ao sujeito x_i da tripla em foco, faz-se também uma busca pelo objeto x_j, sendo que para x_j a busca por c_j é realizada em T (**Def.3**). Com isso, o conjunto C_{x_j} é um conjunto obtido de forma semelhante ao C_{x_i}, porém com sua composição baseada em consultas a T. A partir desse ponto o fluxo principal também segue para a atividade **Buscar Classes equivalentes**. Com os conjuntos C_{x_i} e C_{x_j} formados, o fluxo segue para a atividade **Buscar Predicados relacionados**, onde são utilizadas as funções $domain^{-1}()$ e $range^{-1}()$, definidas em **Def.10** e **Def.11**.

Na atividade seguinte do diagrama da Figura 1, **Identificar potenciais rótulos utilizando RE**, é realizado o processo de *Supervised Relation Extraction* [Han et al. 2019]. É utilizada uma base como fonte para extração das possibilidades de relações. Essa base é composta por textos com assunto no contexto do *dataset* fonte do enriquecimento. Para cada tripla $(x_i, r_k, x_j) \in S$ será realizado um levantamento na base **Dados para extração de relações** constando os predicados disponíveis entre o sujeito (x_i) e o objeto (x_j).

Em seguida, o fluxo unifica-se na atividade **Oferecer para o usuário a oportu-**

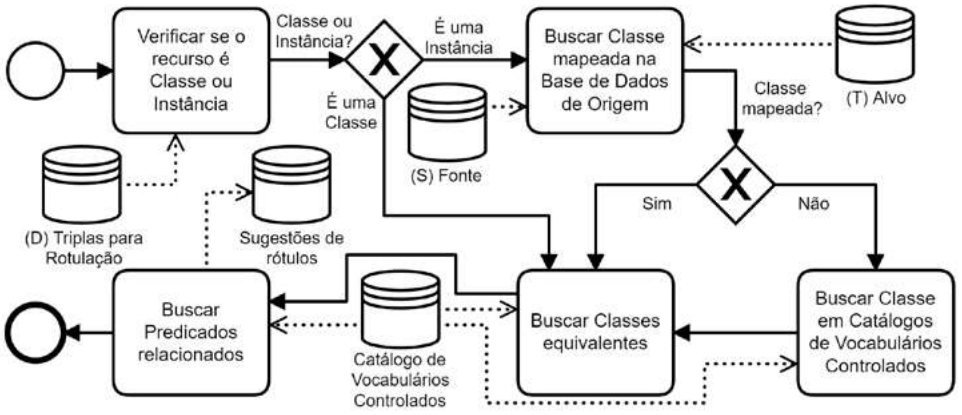

Figura 2. Detalhamento da etapa Consultar Classes e Propriedades correlatas utilizando recursos semânticos em BPMN.

nidade de rotular as ligações, onde são sugeridas ao usuário as opções para nomeação de cada ligação trabalhada. Estarão disponíveis todas as opções levantadas nas etapas anteriores (**Consultar Classes e Propriedades correlatas utilizando recursos semânticos** e **Identificar potenciais rótulos utilizando RE**). A sequência se desdobra em atividades paralelas exclusivas de **Atualizar Predicado** e **Manter Predicado Original**. Nelas, o usuário tem a oportunidade de escolher um rótulo entre os oferecidos ou manter a ligação sem rótulo. A cada escolha de novo predicado, uma nova tripla é então criada e adicionada à base de dados **Triplas Rotuladas**. Após repetir o processo para cada tripla de D o módulo **Reagrupar base** entra em ação, substituindo em S as triplas semanticamente enriquecidas, gerando com isso o dataset S'.

5. PLAIN

Como prova de conceito, foi implementada uma aplicação WEB denominada PLAIN, acrônimo do método *Predicate LAbelINg*. Para o presente trabalho, a aplicação implementa parcialmente o método. Por se tratar de uma linguagem do tipo *software* livre, foi escolhido o PHP[10] com sua biblioteca EasyRdf[11] para o desenvolvimento dessa aplicação. Em especial essa biblioteca foi usada na implementação das etapas **Buscar Classe em Catálogos de Vocabulários Controlados**, **Buscar Classes Equivalentes** e **Buscar Predicados relacionados**. Ela permite a realização de uma série de consultas ao SPARQL Endpoint do Catálogo LOV. O protótipo ainda não tem a opção de consultar diretamente uma base de dados de triplas para rotulação, e necessita que x_i e x_j sejam fornecidos pelo usuário, como sendo parte de uma tripla pobremente rotulada.

A Figura 3 mostra a interface da aplicação em três *frames*. O *frame* A possui dois campos para inserção das classes para consulta (x_i e x_j). O retorno, nos *frames* B e C, são relações de classes equivalentes (C_{x_i} e C_{x_j}, respectivamente), e na parte de baixo dos *frames* estão organizados os predicados associados a essas classes após a aplicação das funções $domain^{-1}()$ e $range^{-1}()$, sobre as classes dos conjuntos C_{x_i} e C_{x_i}.

As etapas **Ler Base de Dados Conectados**, **Consultar Classes e Propriedades correlatas utilizando recursos semânticos**, **Oferecer para o usuário a oportunidade**

[10]https://www.php.net/
[11]http://www.easyrdf.org/

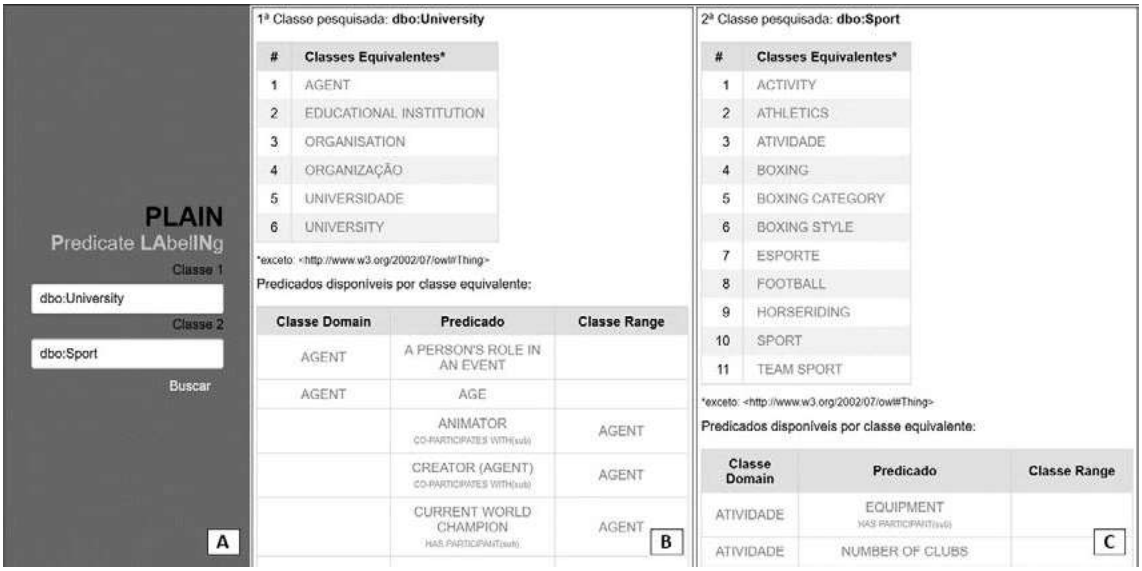

Figura 3. Interface do Protótipo - PLAIN.

de rotular as ligações e **Atualizar Predicado** do *Método Predicate Labeling* estão implementadas na PLAIN por meio da apresentação de uma página estruturada em HTML com o resultado do conjunto de consultas realizadas anteriormente, ou seja, C_{x_i} e C_{x_j}. O usuário tem à sua disposição todo o conjunto de classes e predicados para realizar análise e optar pelo predicado que entender ser o mais adequado para rotular a ligação semântica.

6. Estudo de Caso

Para realização do estudo de caso, foram utilizadas regras geradas como resultados de experimentos de [de Oliveira et al. 2019]. No referido trabalho, foi realizada a mineração de regras de associação de multirrelação em um *dataset* de origem chamado de DtIME (S), que continha informações sobre professores e orientandos da Instituição de Ensino IME. O *dataset* externo utilizado para ampliação foi o DtEsportes (T), que possuía informações sobre a preferência de prática de esportes dos alunos de instituições de ensino. Os autores realizaram experimentos que tiveram como resultados regras de associação entre pares de recursos $(s, t) \in S \times T$. No entanto, a semântica das relações entre os recursos de cada par, não são claras. O objetivo do presente estudo de caso é enriquecer semanticamente essas relações. Por exemplo, a regra de associação *Plays(Vôlei_de_Praia)* $\rightarrow$ *Supervised_By(Work_On(IME))* informa que todos que jogam vôlei de praia são supervisionados por alguém que trabalha na instituição de ensino IME. O que indica que há uma possível relação entre os recursos IME e Vôlei de Praia. Mas qual seria a semântica dessa relação?

O experimento realizado no trabalho supra citado retornou um conjunto de regras como saída. Essas saídas foram entradas para o experimento do presente trabalho. Como os *datasets* S e T são pobres de metadados, para o experimento deste trabalho as classes dos recursos utilizados foram consideradas como inferência do usuário. A partir da regra utilizada, pode-se observar que existe uma relação entre Vôlei de Praia e a instituição de ensino IME. Aplicando a PLAIN para realização da análise, podemos explorar a classe Esporte, de Vôlei de Praia (proveniente de DtEsportes) e a classe Universidade, do IME

(proveniente de DtIME). Neste estudo de caso, foram utilizados os prefixos dbo[12] e dul[13].

O passo seguinte teve como resultado todas as classes equivalentes à classe *dbo:Sport*. A Consulta ao LOV retorna um conjunto de onze classes, conforme observado no *frame* C da Figura 3. Uma delas é a *dbo:Activity*, cujo *label* é Atividade. Entre os predicados mapeados como *Domain* e *Range* dessa classe está o *dbo:equipment* que tem mapeada a sub-propriedade *dul:hasParticipant*, cujo *label* é *has participant*.

Já do ponto de vista da Universidade, classe *dbo:University*, a consulta retorna um conjunto de seis classes (*frame* B da Figura 3). Entre as mapeadas está a classe *dbo:Agent*. Fica explicito que o predicado *dbo:currentWorldChampion* tem essa classe como *Range*. O conjunto de consultas também retorna a informação de que essa é uma sub-propriedade de *has participant*. A sequência de busca do processo fica como apresentado na Figura 4.

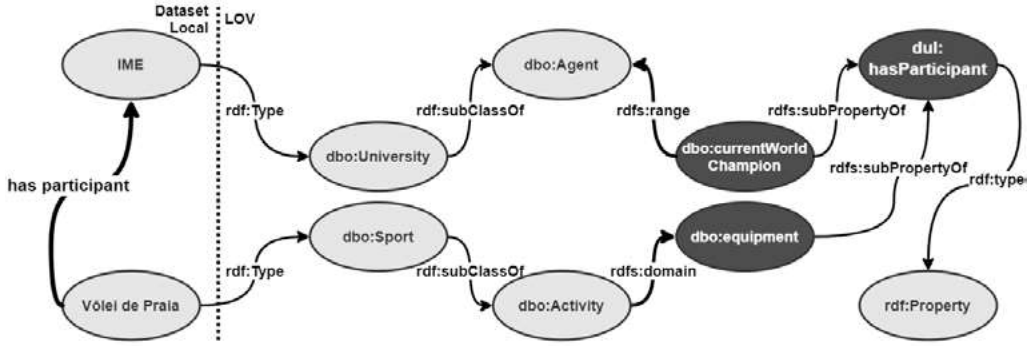

Figura 4. Resultado da sugestão de predicado com uso da PLAIN.

Como resultado desse levantamento feito com o uso da PLAIN, é possível sugerir uma ligação semântica entre Vôlei de Praia e IME utilizando o predicado *dul:hasParticipant*. Além disso, o usuário pode navegar até o local de origem do predicado e explorar mais o seu significado e suas restrições de uso formalizadas. Neste caso, é possível observar no *rdfs:comment* que o predicado *dul:hasParticipant* foi concebido para fazer a ligação entre um objeto e um processo. Dessa maneira, o usuário pode tomar a decisão de aceitar essa sugestão, entendendo que a instituição pode participar de algum processo relacionado com um esporte por um período de tempo, e assim gerar a tripla $< Volei\_de\_Praia\ \ dul{:}\,hasParticipant\ \ IME >$. Outra iniciativa possível seria a de criar uma outra propriedade, como uma sub-propriedade ou propriedade equivalente a *dul:hasParticipant*, incluindo informações mais aderentes ao caso explorado.

Para complementar o estudo de caso, foi realizada a Extração Supervisionada de Relações utilizando a ferramenta OpenNRE[14], conforme [Han et al. 2019], e selecionado modelo BERT. A frase *"sport related to university"* foi inserida na ferramenta que retornou a sugestão de nome *"part of"* para a relação entre *sport* e *university*, com a probabilidade de 80,54%.

Foram realizados outros experimentos com a PLAIN a partir de regras geradas em [de Oliveira et al. 2019] e [de Oliveira et al. 2017]. A Tabela 1 apresenta: as regras selecionadas nesses trabalhos, o sujeito da tripla, o predicado encontrado a partir da navegação

[12]http://dbpedia.org/ontology/
[13]http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#
[14]http://opennre.thunlp.ai/#/sent_re

Tabela 1. Sugestões de rótulos utilizando o método Predicate Labeling.

	Regra Original	Sujeito	Mód. Rec. Sem.	Mód. RE	Objeto
1	*Expedition(Researcher) → Expedition(Researcher)*	*Researcher*	*rel:worksWith*	*said to be the same as* (0,9696)	*Researcher*
2	*Plays (Natacao) → Live_In (RJ)*	*Natacao*	*dul:hasParticipant*	*said to be the same as* (0,6843)	*RJ*
3	*Supervised_By(Maria_Claudia), Study_In(IME) → Plays (Futebol)*	*IME*	*dul:isParticipantIn*	*field of work* (0,9334)	*Futebol*
4	*Supervised_By (Cooperator (Work_On (Patronage(MIT)))) → Study_In (IUT)*	*MIT*	*dul:coparticipates With*	*said to be the same as* (0,9890)	*IUT*
5	*dbo:team(dbr:Brazil_national_under_20 _football_team) → Study_In (IUT)*	*dbr:Brazil _nat_under_20_ft*	*frbr:realizer*	*main subject* (0.4921)	*IUT*

no catálogo de vocabulários (conforme o módulo **Consultar Classes e Propriedades correlatas utilizando recursos semânticos**), a sugestão de predicado e sua probabilidade pela aplicação do RE (conforme o módulo **Identificar potenciais rótulos utilizando RE**) e o objeto da tripla. Em todos os experimentos é possível observar que o retorno dado pela RE é complementar ao obtido com a navegação pelos vocabulários, demonstrando que a combinação das duas abordagens é mais rica.

7. Conclusão

O problema que buscamos tratar neste trabalho é caracterizado pela ausência de semântica observada nos links gerados no processo de enriquecimento de *dataset*. Atualmente as relações encontradas após esse enriquecimento não são nomeadas. Para solucionar esse problema, foi apresentado um método, intitulado como *Predicate Labeling*, que faz uso de ontologias e vocabulários controlados, bem como de técnicas de RE, para favorecer a identificação e concepção dessas relações.

Como contribuição adicional, foi desenvolvida uma aplicação WEB denominada PLAIN que implementa parcialmente a funcionalidade do método proposto. O código fonte está disponível no GitHub[15], e é capaz de realizar uma série de consultas ao SPARQL Endpoint do LOV e organizar os resultados para análise pelo usuário. A PLAIN foi aplicada em um estudo de caso em que foi observado que a rotulação utilizando como base as ligações disponíveis em um repositório de vocabulários é viável, e se mostra como uma solução promissora para o problema levantado por esta pesquisa.

Como trabalhos futuros sugere-se a realização de mais experimentos, incluindo a consulta a outros catálogos de *datasets* e utilização de *datasets* de áreas de conhecimento diversas. A PLAIN também deve evoluir em termos de novas funcionalidades, como por exemplo a materialização das relações que foram encontradas, bem como a implementação o módulo de RE previsto no método.

Referências

Ahmed, A. F., Sherif, M. A., and Ngomo, A.-C. N. (2019). Lsvs: Link specification verbalization and summarization. In *ICANLIS*, pages 66–78. Springer.

Assaf, A., Troncy, R., and Senart, A. (2015). What's up lod cloud? In *ESWC*, pages 247–254. Springer.

[15]https://github.com/rafans222/plain

Athanasiou, S., Giannopoulos, G., Graux, D., Karagiannakis, N., Lehmann, J., Ngomo, A.-C. N., Patroumpas, K., Sherif, M. A., and Skoutas, D. (2019). Big poi data integration with linked data technologies. In *EDBT*, pages 477–488.

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

Bizer, C., Volz, J., Kobilarov, G., and Gaedke, M. (2009). Silk-a link discovery framework for the web of data. In *18th International World Wide Web Conference*, volume 122.

Collovini, S., Gonçalves, P. N., Cavalheiro, G., Santos, J., and Vieira, R. (2020). Relation extraction for competitive intelligence. In *PROPOR*, pages 249–258. Springer.

de Oliveira, F. A., Costa, R., Goldschmidt, R., and Cavalcanti, M. (2019). Multirelation association rule mining on datasets of the web of data. In *SBSI 2019*, page 61. ACM.

de Oliveira, F. A., Martins, Y. C., Rocha, D. S., de Siqueira, M. F., da Silva, L. A. E., Costa, R. L., Goldschmidt, R. R., and Cavalcanti, M. C. (2017). Jabotg: Extending the herbarium dataset frontiers. In *MTSR 2017*, pages 45–53.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*.

Han, X., Gao, T., Yao, Y., Ye, D., Liu, Z., and Sun, M. (2019). Opennre: An open and extensible toolkit for neural relation extraction. In *EMNLP-IJCNLP*, pages 169–174.

Hebeler, J., Fisher, M., Blace, R., and Perez-Lopez, A. (2011). *Semantic web programming*. John Wiley & Sons.

Horrocks, I. (2008). Ontologies and the semantic web. *Comm of the ACM*, 51(12):58–67.

Laufer, C. (2015). Guia de web semântica. *Gov. do Estado de SP e Gov. do Reino Unido*.

Nentwig, M., Hartung, M., Ngonga Ngomo, A.-C., and Rahm, E. (2017). A survey of current link discovery frameworks. *Semantic Web*, 8(3):419–436.

Ngomo, A.-C. N. and Auer, S. (2011). Limes—a time-efficient approach for large-scale link discovery on the web of data. In *IJCAI-11*.

Paris, P.-H. (2018). Assessing the quality of owl: sameas links. In *ESWC*, pages 304–313. Springer.

Ramezani, R., Saraee, M., Nematbakhsh, M. A., et al. (2014). Mrar: mining multi-relation association rules. *Journal of Computing and Security*, 1(2):133–158.

Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *ISWC*, pages 245–260. Springer.

Sherif, M. A., Ngomo, A.-C. N., and Lehmann, J. (2015). Automating rdf dataset transformation and enrichment. In *ESWC*, pages 371–387. Springer.

Studer, R., Benjamins, V. R., and Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197.

Vandenbussche, P.-Y., Atemezing, G. A., Poveda-Villalón, M., and Vatant, B. (2017). Linked open vocabularies (lov): a gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3):437–452.

Yu, L. (2014). *A developer's guide to the SW*. Springer Science & Business Media.