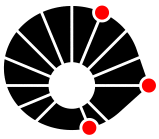

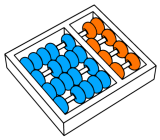

Análise Experimental da Evolução de Links em Dados Interconectados Abertos

Julio Kiyoshi Rodrigues Matsoui *André Gomes Regino*
Julio Cesar dos Reis

Technical Report - IC-20-06 - Relatório Técnico
April - 2020 - Abril

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

Análise Experimental da Evolução de Links em Dados Interconectados Abertos

Julio Kiyoshi Rodrigues Matsoui André Gomes Regino
Julio Cesar dos Reis
Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil.

2020

Resumo

A descrição de dados abertos em formato padrão sobre entidades na Web tem tido uma grande adoção nos últimos anos. O principal formato explorado na definição de elementos de dados é o *Resource Description Framework* (RDF). O número de repositórios de dados nesse formato provendo dados interconectados entre diferentes fontes se tornou numeroso. O grande volume de dados exige cada vez mais métodos e ferramentas automáticas para efetuar análises e correções sobre os dados. Em particular, dados interconectados no contexto da Web Semântica tendem a ser dinâmicos. Novas versões de conceitos, suas relações e instâncias são redefinidos ao longo do tempo e podem alterar o significado e propriedades das entidades. Logo, tornam-se necessárias técnicas para se entender a evolução de dados interconectados e como tal evolução influencia em interconexões estabelecidas entre bases distintas. Este relatório apresenta e discute um conjunto de análises experimentais para entender a evolução de links entre repositórios de dados descritos em RDF. Desenvolvemos uma ferramenta de análise automatizada das mudanças entre versões de uma base de conhecimento. Nossas análises investigam como essas mudanças afetam links pré-estabelecidos. Resultados obtidos indicam que as mudanças que mais ocorrem são mudanças complexas, triplas as quais sofrem algum tipo de modificação, mas especificamente modificação de um predicado ou objeto da tripla sem modificação do link relacionado ao sujeito.

1 Introdução

Uma característica fundamental da Web reside no fato da existência de links que apontam de um documento para outro. No contexto da Web Semântica, esses links

expressam interconexão entre elementos de dados estruturados com semântica explicitamente definida. Assumindo que as propriedades das entidades descritas em repositórios de dados expressos em RDF[1] podem sofrer modificações, esta podem impactar em interconexões (links) já estabelecidas. O grande volume de dados atualmente nos repositórios de dados da Web Semântica impede que correções manuais sejam efetuadas em situações em que modificações invalidam conexões previamente definidas.

Estudos detalhados são necessários para determinar os tipos de modificações possíveis nesses repositórios e como eles influenciam no comportamento das interconexões existentes. Ainda não se tem conhecimento se, e quais, modificações nas bases encadeiam quais modificações nas interconexões. Se a manutenção das interconexões não é desenvolvida ao longo do tempo, a utilidade das bases de conhecimento interconectadas pode ser reduzida, pois não se apresenta precisão sobre a validade dos links existentes.

Como em várias áreas do conhecimento, dados abertos interconectados (no inglês, *Linked Open Data - LOD*) evoluem com o passar do tempo em versões da mesma base. Nesse sentido, de uma versão do dataset - nome genérico para os diversos tipos de bases de dados conhecidos - para outra podem ocorrer mudanças, uma vez que os dados são dinâmicos. Esse fato pode trazer inconsistências, como por exemplo, o surgimento de um link "quebrado" entre elementos de dados. De maneira mais contextualizada, aplicações, como na área médica, que utilizam de dados externos para o apoio ao diagnóstico, necessitam manter as bases de dados atualizadas. Potenciais mudanças nos conjuntos de dados podem acarretar efeitos indesejados na aplicação de software.

Neste relatório técnico descrevemos análises experimentais realizadas a fim de se entender como modificações ocorrem de uma versão de uma base de conhecimento para outra. Tendo em vista a importância do maior entendimento sobre o assunto, é necessário um estudo mais profundo sobre o problema das mudanças impactando os links entre bases existentes. Neste relatório, propomos a definição das seguintes análises experimentais: (1) efeito da remoção nas triplas influenciando os respectivos links; (2) adição de uma nova instância e com adição ou não de um link; (3) modificações ocorridas nas triplas, podendo adicionar, remover ou modificar um link; (4) modificação na tripla sem modificar o link.

Essas análises devem permitir entender e prever as consequências das mudanças efetuadas, sendo elas mudanças simples e complexas. Para tal, é necessário entender os fatores que levam às mudanças, possibilitando a detecção automática dessas nas bases, para assim, adaptar-se prevendo as mudanças nos links. Pretende-se assim automatizar as análises propostas de modo a conduzir experimentos extensivos com diversos conjunto de dados na LOD.

[1] https://www.w3.org/RDF/

As seções seguintes deste relatório estão organizadas da seguinte maneira: Seção 2 efetua uma síntese da revisão da literatura apresentado estudos relacionados à esta investigação; Seção 3 formaliza conceitos no contexto de dados abertos conectados, apresenta as definições das análises propostas e as implementações; a Seção 4 mostra os resultados obtidos das análises propostas na Seção 3, enquanto a Seção 5, discute sobre os resultados obtidos; por fim a Seção 6 apresenta a conclusão do trabalho.

2 Revisão da Literatura

Apresentamos e discutimos propostas existentes na literatura para o problema das mudanças dos dados interconectados. Papastefanatos *et al.* (7) estudou uma modelagem sobre mudança de dados utilizando evo-graph. Evo-graph é um modelo que visa compreender como ocorre a evolução dos dados em formato de grafos. Para tanto criou-se uma ferramenta, chamada de C2D, que efetua as análises de mudanças de modo automático. Os resultados foram as definições de mudanças de uma versão para as outras baseando-se no modelo proposto e a aplicação desse conceito na ferramenta C2D.

Em outro estudo sobre as mudanças dos dados interligados, tipos de mudanças complexas são definidos para mudanças das bases em RDF. Para isso, Galani *et al.* (3) definiram tipos de mudanças simples e complexas. Adicionalmente, os autores propuseram como detectar mudanças entre duas versões de uma mesma base. Considerando uma formalização matemática para os conceitos apresentados em (3), o paper de Galani *et al.* (2) trata as mudanças complexas com maior refinamento matemático e de maneira que simplifique os conceitos de mudanças simples e complexas; O paper também apresenta um algoritmo para detectar mudanças complexas.

Tobias Käfer *et al.* (5) propuseram modelar *Linked Data* usando RDF para que as interdependências dos dados sejam investigadas mais facilmente com o uso de SPARQL[2]. Os autores demonstraram que com a otimização das consultas, ou seja, otimizando as triplas e usando o corpo em RDF ao invés de HTTP, foi possível encurtar o tempo do resultado das consultas nos conjunto de dados interligados.

Uma outra perspectiva das mudanças em datasets é analisar uma maneira de arquivar datasets abertos e interligados, sabendo que esses evoluem tanto na semântica quanto estruturalmente. Nessa perspectiva, Meimaris *et al.* (6) propuseram um *framework* para gerenciar a evolução de recursos RDF. Os autores sugeriram um modelo para evolução dos dados interconectados, propondo um espaço 2x2 no qual os objetos são separados pela dependência temporal assim como a evolução da semântica. Para isso, usou-se datasets com identificadores diacrônicos, vinculados às suas versões temporais e são calculadas as mudanças simples e complexas entre as versões do dataset.

[2]https://www.w3.org/TR/rdf-sparql-query/

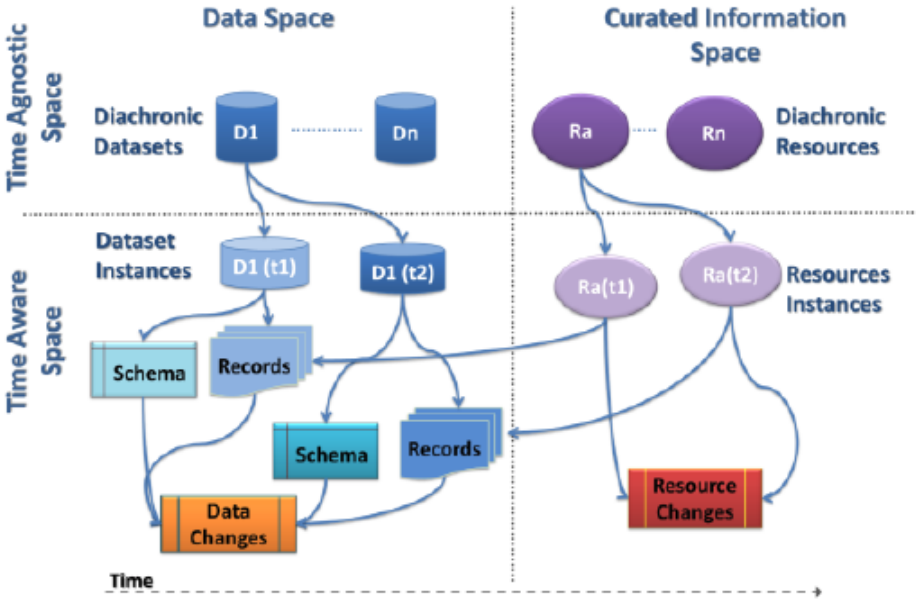

Figura 1: Modelo do *dataspace*. Retirado de (6)

A Figura 1 apresenta a proposta de Meimaris *et al.* (6), que apresenta o calculo de mudanças no dataset.

Tobias (5) abordou o problema de mudanças nas bases RDF visando uma maior compreensão e análise da dinâmica dos datasets no LOD. O uso de SPARQL foi necessário para efetuar consultas e, assim, observar a dinâmica dos dados ligados. Para otimizar as consultas em SPARQL, os autores propuseram a reordenação ou reformulação das consultas. Como uma solução alternativa, recomendou-se reduzir o número de triplas a serem consultadas, através do uso de um "*blank node*" ou URI, pois na presença de nós vazios, o grafo de RDF pode usar diferentes identificadores de "*blank node*" para representar os mesmos dados.

Nessa abordagem, foi detectado isomorfismo entre grafos de RDF e, com isso, feito uma substituição dos "*blank nodes*" por URIs usando *hash* baseado na skolemização, proposta por Hogan (4). Dessa forma, foi possível conferir se uma tripla com um "*blank node*" procedente de um tempo é a mesma que uma tripla com um "'*blank node*" em um outro tempo, usando a igualdade RDF URI. Assim a otimização da *query*, é possivel através da introdução de "*blank nodes*" ou URI para o corpo de resposta HTTP e usar triplas com o predicado *rdfs:member* para conectar ao corpo, HTTP, das declarações no grafo RDF.

Estudos sobre a diferença entre versões de bases RDF também têm sido conduzidos. Por exemplo, Berners-Lee & Connolly (1) trataram a diferença entre duas versões de ontologias usando uma extensão de RDF, Notation3 (n3). Utilizaram-se

das definições de RDF (grafo orientado), percorrendo o grafo entre as versões com a finalidade de obter o maior subgrafo comum entre elas, obtendo então o cálculo dos *Diffs* entre versões.

Para o problema anterior, foi proposto um arquivo de correção, sendo este arquivo de correção uma nova ontologia de duas ou três propriedades novas, em relação à atual, o qual utiliza a notação semântica e sintática da Notation3 (extensão do RDF). Esse formato pode ser gerado por algoritmos de busca de diferenças.

Um método de implementar os cálculos e os estudos das diferenças entre as versões de bases em RDF foi proposta pelo projeto *Diachron*, através da ferramenta D2V[3], feita em um dataset específico[4]. Com o D2V é possível calcular as mudanças simples e complexas que ocorrem entre as versões do dataset. As mudanças complexas podem ser redefinidas de acordo com suas especificações que buscam, entre as mudanças, por exemplo, nomear a mudança, priorizar e atribuir uma assinatura de mudança simples, e, por fim, inserir os parâmetros da mudança e a descrição. Contudo, o D2V foi criado para datasets específicos, o que representa uma limitação, já que isso impossibilita o cálculo de mudanças ocorridas entre quaisquer datasets.

Os trabalhos na literatura apresentaram diversas contribuições para o gerenciamento e análise de conjunto de dados em RDF. Contudo, a definição e a condução de análises experimentais sobre os efeitos de mudanças sobre links entre diferentes bases não foi estudado em profundidade. Logo, a área de estudo merece investigações que permitam identificar requisitos para a construção de mecanismos automatizados no gerenciamento da evolução de bases RDF interconectadas.

3 Análises Experimentais para Entender Evolução de Links

Esta investigação visa estudar os efeitos das mudanças ocorridas nas instâncias de dados RDF e seus impactos em links previamente estabelecidos. Dessa forma, definimos uma série de análises para averiguar como as mudanças em um dataset influenciam os links das instâncias. Mais precisamente, este trabalho objetiva estudar como os links são afetados quando ocorre uma mudança atingindo as instâncias de uma versão para outra de um dataset. A Figura 2 apresenta a problemática do estudo. Nesse contexto, assumimos que a base R^{Tj} da Figura 2 não se altera com o tempo, ou seja, ela é estática.

Na Figura 2, por exemplo, em um tempo j há um dataset (R_1^j) na qual uma instância está ligada a outra instância; após um Δj houve uma nova versão do dataset (R_1^{j+1}) e ocorreu a remoção do recurso r_a; isso pode afetar negativamente uma ligação

[3]https://github.com/hrysakis/d2v_change_detection_tool

[4]https://www.ebi.ac.uk/efo/

existente.

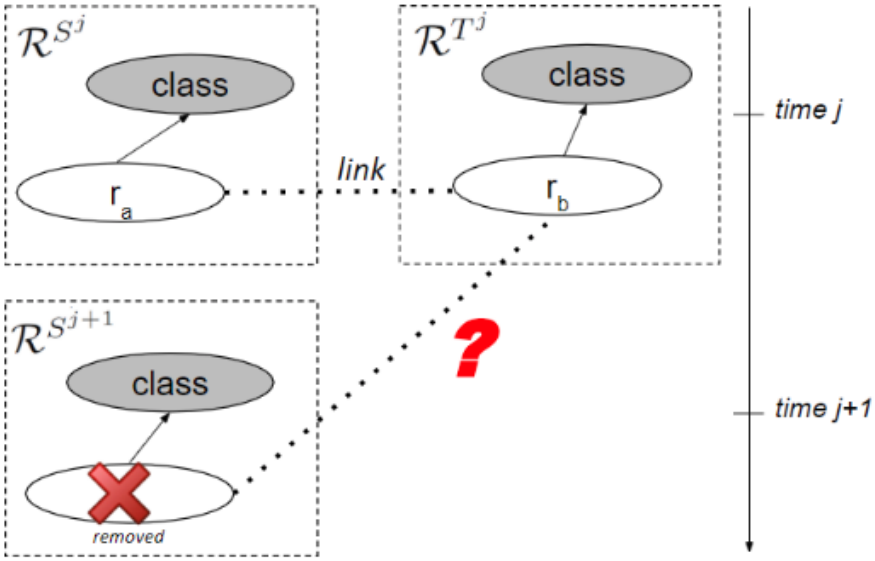

Figura 2: Evolução das instâncias em um Δ

Elaboramos um conjunto de definições para fundamentar as análises propostas.

Definição 1 (RDF): RDF (*Resource Description Framework*) é um modelo padrão para intercâmbio de informação que utiliza-se de grafos rotulados e direcionados para representar dados na Web. Esse modelo permite que dados estruturados e semiestruturados sejam combinados, espostos e compartilhados entre diversas aplicações. Esse modelo de grafos fundamenta a sintaxe e a semântica do SPARQL query language[5] para RDF, uma linguagem para consulta em bases RDF, em que as consultas expressam padrões e percorrem caminhos em grafos para se obter resultados.

Definição 2 (Triplas): Os elementos de dados em RDF são definidos como triplas, uma tripla pode ser representada por R^S, sendo esta formada por um conjunto de três elementos (s, p, o), na qual s é o sujeito, p é o predicado e o é o objeto; Os elementos são expressos por endereços na rede, ou seja, são URIs (Uniform Resource Identifier). A tripla é definida quando o sujeito e o objeto têm o mesmo domínio (estrutura do URL). Um exemplo de tripla:

- s = "http://dbpedia.org/resource/Actrius";
- p = "http://dbpedia.org/ontology/country";
- o = "http://dbpedia.org/resource/Spain";

[5]https://www.w3.org/TR/rdf-sparql-query/

Definição 3 (Link): um link é constituído por uma tripla "externa", cujo sujeito se encontra em uma base RDF e o objeto em outra base RDF, ou seja, o domínio do sujeito é diferente do domínio do objeto. Assim ocorre um link entre a base do domínio do sujeito com a base do domínio do objeto, feito por um predicado particular. Exemplos desses predicados são: "sameAs", "exactMatch", "closeMatch", entre outros. Um exemplo de link:

- s = "http://aims.fao.org/aos/agrovoc/c_4788" ;
- p = "http://www.w3.org/2004/02/skos/core#exactMatch";
- o = "http://d-nb.info/gnd/4038971-6";

É válido ressaltar que o domínio do sujeito *"aims.fao.org"* é diferente do domínio do objeto *"d-nb.info"*, caracterizando assim a presença de um link.

Definição 4 (Dataset): É um conjunto de triplas RDF referentes a um determinado domínio. O dataset em um tempo j, pode ser definido por R^j que é o conjunto das triplas e dos links.

Definição 5 (*Diff*): *Diff* pode ser caracterizado como $\Delta(R^j, R^{j+1}) = R^j - R^{j+1}$, ou seja, a computação da diferença entre as triplas desses datasets. Nesse procedimento, removemos todas as instâncias que são iguais entre R^j e R^{j+1}, ficando apenas com as instâncias diferentes pertencentes a R^{j+1}, na qual o dataset no tempo j é R^j é no tempo $j+1$ é R^{j+1}.

Definição 6 (Mudanças): As mudanças ocorrem entre duas versões de um mesmo dataset, podendo ser analisadas de maneira mais simples (*fine-grained*) e de modo mais complexo (*coarse-grained*).

- **Definição 6.1 (Mudanças simples:)** Seja R^j uma versão do dataset e R^{j+1} a nova versão do dataset, denotamos um conjunto de triplas formadas por (Δ_1, Δ_2) com o conjunto das mudanças simples, sendo que $\Delta_1 = (R^j - R^{j+1})$ e $\Delta_2 = (R^{j+1} - R^j)$.
  - **Definição 6.1.1 (Mudança simples de adição):** Na mudança simples de adição ocorre uma adição na lista de instância do dataset R^{j+1} em relação a lista de instâncias da versão anterior R^j, ou seja, quando Δ_2, definição 6.1, é uma lista não vazia de instâncias.
  - **Definição 6.1.2 (Mudança simples de remoção):** Na mudança simples de remoção ocorre uma remoção da lista de instâncias do dataset R^{j+1} em relação a versão anterior R^j.
- **Definição 6.2 (Mudança complexa):** Mudanças complexas consistem de uma lista não vazia de mudanças simples já definidas, e um conjunto de restrições sobre essas mudanças. As restrições podem filtrar valores de parâmetro,

expressar pré ou pós condições, relacionar mudanças de parâmetro, representar restrições de cardinalidade (por exemplo, deve haver pelo menos uma mudança específica de um tipo); e permitir ou não sobreposições entre as alterações.

3.1 Definição das Análises

Definimos as seguintes análises experimentais para averiguar a evolução de links em diferentes situações de mudanças:

a) Remoção de uma instância acarretando em uma remoção de um link.

A remoção de uma instância com remoção de um link é apresentado pela Figura 3 (a); no tempo j em um dataset R^{Sj} há um link para o dataset R^{Tj} conectando os recursos r_a e r_b. No tempo $j+1$, ocorre uma remoção do sujeito r_a e visamos detectar ser há remoção do link.

b) Remoção de uma instância sem a remoção de link.

A remoção de uma instância sem remoção de um link é representada pela Figura 3 (b); no tempo j em um dataset R^{Sj} há um link para o dataset R^{Tj} conectando os recursos r_a e r_b. No tempo $j+1$, ocorre uma remoção do sujeito r_a e visamos detectar se o link não é removido.

c) Adição de uma nova instância sem a criação de um link.

A Figura 3 (c) apresenta a situação de adição de uma nova instância sem a criação de uma link. Uma nova tripla é adicionada em R^{Sj+1} mas não há ocorrência de um novo link.

d) Adição de uma nova instância acarretando a criação de um link.

A Figura 3 (d) apresenta a situação de adição de uma nova instância com a criação de uma link. Uma nova tripla é adicionada em R^{Sj+1} e detectamos que há ocorrência de um novo link entre tal novo recurso de R^{Sj+1} com R^{Tj}.

e) Modificação de um predicado ou objeto da tripla acarretando uma adição de um link relacionado ao sujeito da tripla.

A Figura 4 (e) apresenta este caso. Um tripla t em R^{Sj} é modificada (i.e., tem seu predicado ou objeto modificado). Essa tripla não está relacionada com nenhum link para outro dataset. Então, após a evolução, se detecta que um novo link é criado e relacionado ao sujeito da tripla t.

f) Modificação de um predicado ou objeto da tripla acarretando uma remoção de um link relacionado ao sujeito da tripla.

A Figura 4 (f) apresenta este caso. Um tripla t em R^{Sj} é modificada (*i.e.*, tem seu predicado ou objeto modificado). Essa tripla está relacionada com um link para outro dataset no tempo j. Após a evolução, se detecta que tal link é removido.

g) Modificação de um predicado ou objeto da tripla com a modificação do link relacionado ao sujeito.

A Figura 4 (g) apresenta este caso. Um tripla t em R^{Sj} é modificada (*i.e.*, tem seu predicado ou objeto modificado). Essa tripla está relacionada com um link para

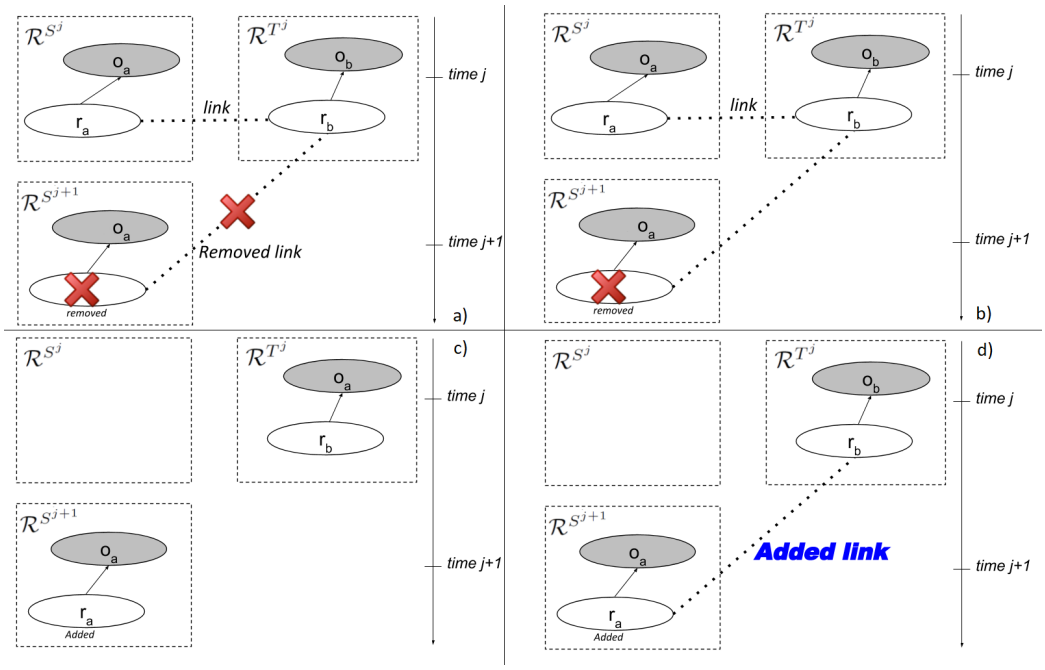

Figura 3: Análise da evolução dos links baseado em remoção e adição de triplas.

outro dataset no tempo j. Após a evolução, se detecta que tal link é modificado, identificando que o predicado p_a ou o objeto o_a do link é diferente.

h) Modificação de um predicado ou objeto da tripla sem modificar o link relacionado ao sujeito.

A Figura 4 (h) apresenta este caso. Um tripla t em R^{S^j} é modificada (*i.e.*, tem seu predicado ou objeto modificado). Essa tripla está relacionada com um link para outro dataset no tempo j. Após a evolução, se detecta que tal link se mantém inalterado, ou seja, ele continua relacionado os mesmos recursos, embora a tripla em $R^{S^{j+1}}$ tenha sofrido alterações.

3.2 Implementação das Análises

Desenvolvemos uma ferramenta de software[6] para implementar as análises definidas.

Figura 5 apresenta a ferramenta implementada. Após a escolha do dataset desenvolvemos a funcionalidade de leitura dos mesmos, proposta através de algoritmo simplificado de leitura de arquivos. Posteriormente, deve-se calcular o *Diff*, segundo a Definição 5 (cf. Seção 3), através de uma comparação entre arquivos. Como estamos usando um formato turtle (*.ttl*) - em que cada linha consiste em uma tripla - é possível tokenizar cada um dos três itens de cada linha em sujeito, predicado e objeto com o

[6]https://gitlab.ic.unicamp.br/jreis/evoLOD-analysis.git

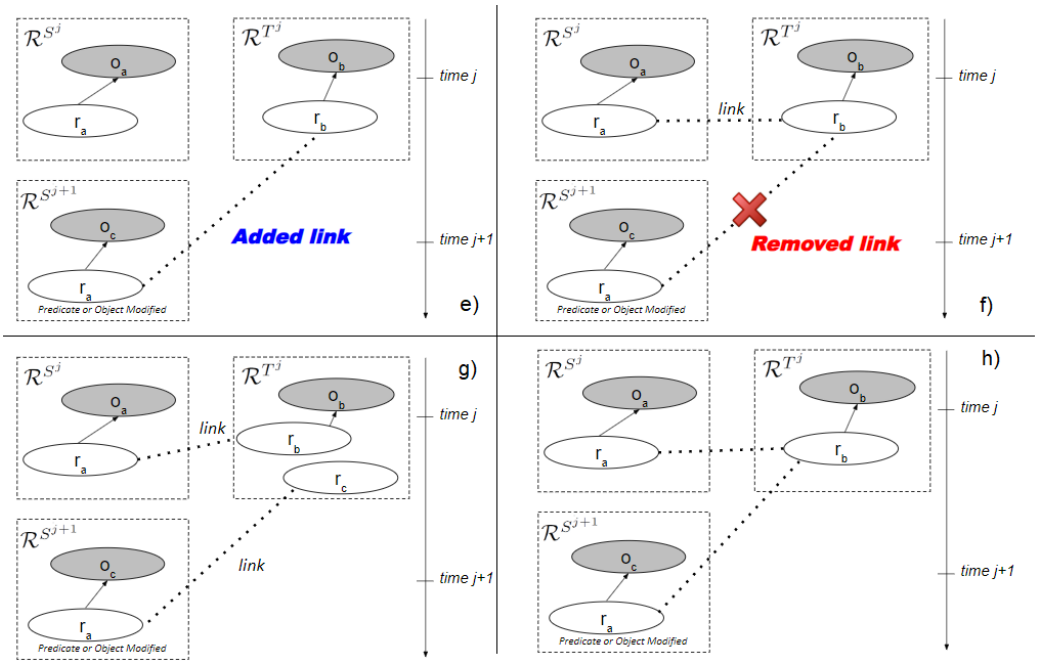

Figura 4: Análise da evolução dos links baseados nas modificações das triplas.

uso da função *split()*, que se encontra na classe String. Dessa forma, comparamos os sujeitos, predicados e objetos entre o dataset antigo e novo.

A implementação foi separada em três partes principais: adição, remoção e modificação. Cada uma dessas funções gerou pelo menos uma listagem de triplas e links, diferenciadas pelo domínio do objeto: caso o domínio seja o mesmo do que o dataset em questão, é identificado como tripla e caso o domínio seja diferente, pertencente a um dataset externo, por consequência é identificada como um link.

O *Diff* da adição é calculado através de uma comparação entre as versões do datasets R^{j+1} e R^{j} , nessa respectiva ordem. Assim é possível verificar as adições que ocorreram de uma versão para outra. Já o *Diff* da remoção é possível ser calculado através da comparação entre o dataset R^{j} e R^{j+1}, nessa respectiva ordem, pois é possível verificar o que está contido no dataset R^{j}, mas não estará contido em R^{j+1}, caracterizando uma remoção.

O *Diff* da modificação é realizado através de uma comparação de equivalência entre o sujeito das versões do dataset, porém com uma mudança no predicado ou objeto da tripla. O cálculo do *Diff* é importante, pois reduz drasticamente o tamanho do dataset e seleciona apenas as triplas e os links que sofreram algum tipo de mudança.

Por último, com os *Diff*s implementados o algoritmo procede para a execução das análises propostas na Seção 3.1. Para isto, a ferramenta implementa um conjunto de algoritmos que caracterizam diferentes módulos de implementação. Para implementar

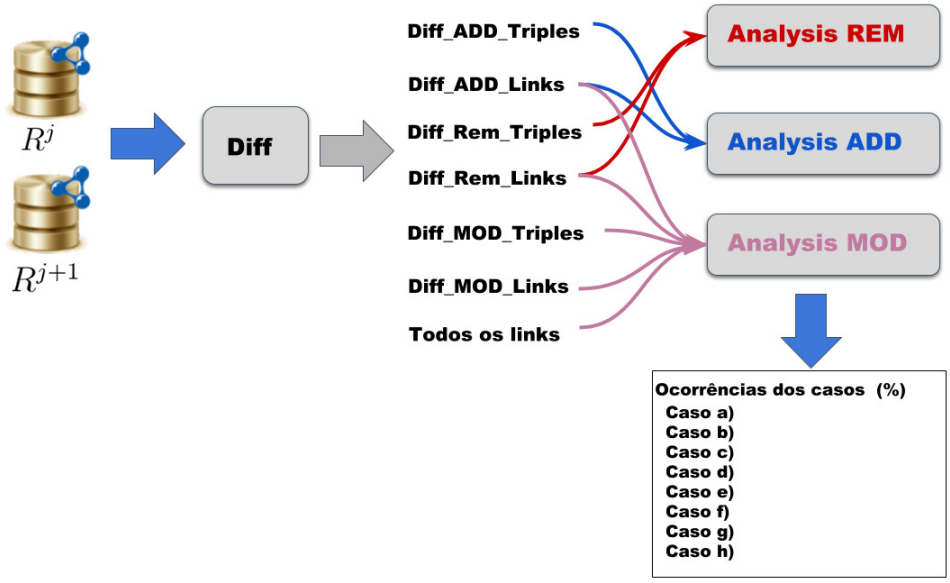

Figura 5: Ferramenta para computar as análises propostas

esses módulos utiliza-se de arquivos que contêm os resultados dos *Diff*s da adição, remoção e da modificação. Segue a baixo a implementação dos módulos:

O algoritmo 1 da análise da remoção tem como entradas o *Diff* da remoção contendo os links e as triplas removidas. Primeiramente, usa-se a função 'split' nos *Diff*s triplas e links nas linhas 6 e 9 respectivamente para possibilitar a comparação entre sujeito, predicado e objeto de ambos *Diff*Triples e *Diff*Links. Dentro do laço mais interno na linha 13 compara-se os sujeitos, que se dá através de *triplesIN[0]* == *triplesEX[0]*, ou seja, se temos uma igualdade entre os sujeitos das triplas com os sujeitos dos links, temos o caso *a*, o qual é adicionado em uma lista na linha 14 que posteriormente é armazenada em um arquivo de resultado das análises de remoção. A variável *temp*, ou temporária, é usada apenas para fazer um controle de triplas para que estas não sejam adicionadas duplamente posteriormente. Na próxima etapa, as triplas que não participam do caso *a* serão adicionadas no caso *b* na linha 19.

O algoritmo 2 da análise da adição é similar ao da remoção, com a diferença nos dados usados como entrada, que no caso da adição, usa-se os arquivos contendo as triplas e os links adicionados geradas pelo módulo do *Diff* da adição.

O algoritmo 3 da análise da modificação tem como entradas os links do *Diff* da remoção, os links do *Diff* da adição, os links e as triplas gerados pelo *Diff* da modificação, representados entre as linhas com o prefixo 'require'. Similar aos algoritmos propostos anteriormente, é necessário realizar o split das triplas e dos links - linhas 9, 12, 15 e 18. Na próxima etapa, percorre-se as listas geradas (links modificados,

Algorithm 1: Algoritmo para computar as análises de remoção.

Require: $DiffRemLinksFile$, $DiffRemTriplesFile$

1: $Rem_{Links} \leftarrow DiffRemLinksFile$
2: $Rem_{Triples} \leftarrow DiffRemTriplesFile$
3: $case_a \leftarrow List$
4: $case_b \leftarrow List$
5: **for all** $Rem_{Triples}$ **do**
6: $Triples_{IN} = Split(Rem_{Triples})$
7: **end for**
8: **for all** Rem_{Links} **do**
9: $Triples_{EX} = Split(Rem_{Links})$
10: **end for**
11: **for all** $Triples_{IN}$ **do**
12: **for all** $Triples_{EX}$ **do**
13: **if** $Triples_{IN}[0] == Triples_{EX}[0]$ **then**
14: $case_a.add = Triples_{IN}$
15: **else**
$temp++$
16: **end if**
17: **end for**
18: **if** $temp == Rem_{Links}.size$ **then**
19: $case_b.add = Triples_{EX}$
20: **end if**
21: **end for**

triplas adicionadas, removidas e modificadas) e realiza-se algumas comparações: na linha 22 a primeira comparação entre sujeitos para verificar se há o caso onde temos uma modificação da tripla seguido de uma adição de um link; na linha 27 a segunda comparação apuramos se temos uma modificação na tripla seguido de uma remoção de um link; na linha 32 a última comparação se dá para averiguar o caso da modificação da tripla com a modificação do link. Por fim, para o caso h, subtraímos todos os casos de modificação do total de triplas presentes no *Diff* da modificação, assim temos o caso da modificação das triplas sem ocorrer modificação dos links.

Algorithm 2: Algoritmo para computar as análises de adição.

Require: $DiffAddLinksFile$, $DiffAddTriplesFile$

1: $Add_{Links} \leftarrow DiffAddLinksFile$
2: $Add_{Triples} \leftarrow DiffAddTriplesFile$
3: $case_c \leftarrow List$
4: $case_d \leftarrow List$
5: **for all** $Add_{Triples}$ **do**
6: $Triples_{IN} = Split(Add_{Triples})$
7: **end for**
8: **for all** Add_{Links} **do**
9: $Triples_{EX} = Split(Add_{Links})$
10: **end for**
11: **for all** $Triples_{IN}$ **do**
12: **for all** $Triples_{EX}$ **do**
13: **if** $Triples_{IN}[0] == Triples_{EX}[0]$ **then**
14: $case_d.add = Triples_{IN}$
15: **else**
$temp++$
16: **end if**
17: **end for**
18: **if** $temp == Rem_{EX}.size$ **then**
19: $case_c.add = Triples_{EX}$
20: **end if**
21: **end for**

4 Avaliação Experimental

Apresentamos os resultados experimentais das análises conduzidas. Esta seção esta organizado da seguinte forma: a subseção 4.1 apresenta os datasets usados e o protocolo experimental; as subseções subsequentes descrevem os resultados obtidos organizados em três grupos principais: Subseção 4.2 apresenta os resultados da forma como a remoção afeta os links; Subseção 4.3 apresenta os resultados do efeito da adição nos links; Subseção 4.4 relata os efeitos da modificação nos links.

4.1 Datasets e protocolo experimental

A escolha do conjunto de dados foi feita a partir de dois critérios: ele deve conter tanto links quanto triplas e deve haver mais de uma versão disponível para comparação. Em uma análise inicial, o dataset adequado que cumpri os critérios apresentados foi

Algorithm 3: Algoritmo para computar as análises de modificação.

Require: $DiffModLinksFile$,$DiffModTriplesFile$, $DiffRemLinksFile$, $DiffAddLinksFile$

1: $Add_{Links} \leftarrow DiffAddLinksFile$
2: $Rem_{Links} \leftarrow DiffRemLinksFile$
3: $Mod_{Links} \leftarrow DiffModLinksFile$
4: $Mod_{Triples} \leftarrow DiffModTriplesFile$
5: $case_e \leftarrow List$
6: $case_f \leftarrow List$
7: $case_g \leftarrow List$
8: **for all** Add_{Links} **do**
9: $TriplesAdd_{EX} = Split(Add_{Links})$
10: **end for**
11: **for all** Rem_{Links} **do**
12: $TriplesRem_{EX} = Split(Rem_{Links})$
13: **end for**
14: **for all** Mod_{Links} **do**
15: $TriplesMod_{EX} = Split(Mod_{Links})$
16: **end for**
17: **for all** $Mod_{Triples}$ **do**
18: $TriplesMod_{IN} = Split(Mod_{Triples})$
19: **end for**
20: **for all** $TriplesMod_{IN}$ **do**
21: **for all** $TriplesAdd_{EX}$ **do**
22: **if** $TriplesMod_{IN}[0] == TriplesAdd_{EX}[0]$ **then**
23: $case_e.add = Mod_{Triples}$
24: **end if**
25: **end for**
26: **for all** $TriplesRem_{EX}$ **do**
27: **if** $TriplesMod_{IN}[0] == TriplesRem_{EX}[0]$ **then**
28: $case_f.add = Mod_{Triples}$
29: **end if**
30: **end for**
31: **for all** $TriplesMod_{EX}$ **do**
32: **if** $TriplesMod_{IN}[0] == TriplesMod_{EX}[0]$ **then**
33: $case_g.add = Mod_{Triples}$
34: **end if**
35: **end for**
36: **end for**

o *Agrovoc*[7]. Após a escolha do dataset, investigamos as versões utilizadas para o cálculo do *Diff* e das análises propostas. Dessa maneira, escolhemos as versões: abril de 2018 (R^j) e abril de 2019 (R^{j+1}). Identificamos as seguintes características sobre a computação da diferença entre essas versões:

- Total de triplas e links do dataset R^j é de 4.254.655 e da nova versão, R^{j+1}, apresentou um total de 4.540.205 triplas e links.
- O Diff, $\Delta(R^j; R^{j+1})$, detectou: 202 remoções, 6.990 adições e 528.275 modificações.
- Os predicados dos links no dataset, estão dividas em *'closeMatch', 'exactMatch', 'narrowMatch', 'broadMatch'*.

4.2 Resultados: Efeito da remoção

A Tabela 1 apresenta os resultados para os casos (a) e (b) da seção 3.1, em que R:1 representa o caso (a) e R:2 o caso (b) (*cf.* subseção 3).

Resultados das análises das remoções			
Id	Tipo	R:1	R:2
(a)	Remoção de tripla com remoção de link	3	-
(b)	Remoção de tripla sem remoção de link	-	75
(c)	Total de triplas e links removidos	202	202
(d)	Total de triplas e links no dataset no tempo j	4.254.655	4.254.655
(e)	Total de triplas e links no dataset no tempo j+1	4.540.205	4.540.205
(f)	Porcentagem (a ou b)/(c)	1.49%	37.13%
(g)	Porcentagem (a ou b)/(e)	$< 0.01\%$	$< 0.01\%$

No caso (a) detectou-se 78 remoções de triplas. O total de triplas e links presentes em R^j que foram removidas em R^{j+1} é 202, o que nos dá um total de 124 links removidos. Também foi detectado que em apenas 3 casos ocorreram a remoção de tripla seguido de remoção de link, o que representa menos de 0.01% dos eventos ocorridos no dataset, id (g).

Já no caso (b) foi encontrado um total de 75 ocorrências de remoção de triplas sem a remoção de link, o que representa 37.13% das ocorrências de remoções, id (f), e representa menos de 0.01% sobre o dataset como um todo, id (g).

4.3 Resultados: Efeito da adição

Na Tabela 2 está representado os casos (c) e (d) da seção 3.1. Na qual A:1 e A:2, representam os casos (d) e (c) respectivamente.

[7]http://aims.fao.org/agrovoc/releases

Resultado das análises da adição			
Id	Tipo	A:1	A:2
(a)	Adição de tripla com adição de link	6909	-
(b)	Adição de tripla sem adição de link	-	29
(c)	Total de triplas e links adicionados	6.990	6.990
(d)	Total de triplas e links no dataset no tempo j	4.254.655	4.254.655
(e)	Total de triplas e links no dataset no tempo j+1	4.540.205	4.540.205
(f)	Porcentagem (a ou b)/(c)	98.84%	0.41%
(g)	Porcentagem (a ou b)/(e)	0.15%	$< 0.01\%$

No caso (d), temos um total de 6909 triplas adicionadas seguido de uma adição de um link. Isto representa no total do dataset 0.15% ,id (g), porém quando comparamos em relação às adições o caso (d) se torna mais significativo com uma porcentagem de 98,84%, id (f). No caso (e), temos um total de 29 triplas adicionadas sem a adição de link. O qual representa no total do dataset menos de 0.01% , id(g) , dele, quando comparado somente com as adições ocorridas é representado por 0.41%, id (f).

4.4 Resultados: Efeito da modificação

A Tabela 3 representa os casos (e), (f), (g) e (h) da seção 3.1. Na qual M:1, M: 2, M:3 e M:4 representam o casos (e), (f), (g) e (h), respectivamente.

Resultados das análises da modificação.					
Id	Tipo	M:1	M:2	M:3	M:4
(a)	Mod. de tripla com Adi. de link	0	-	-	-
(b)	Mod. de tripla com Rem. de link	-	245	-	-
(c)	Mod. de tripla com Mod. de link	-	-	21.899	-
(d)	Mod. de tripla sem Mod. de link	-	-	-	474.253
(e)	Total de triplas e links Modificadas	528.275	528.275	528.275	528.275
(f)	Total de triplas e links no dataset no tempo j	4.254.655	4.254.655	4.254.655	4.254.655
(g)	Total de triplas e links no dataset no tempo j+1	4.540.205	4.540.205	4.540.205	4.540.205
(h)	Porcentagem (a ou b ou c ou d)/(e)	0.00%	0.04%	4.14%	95.82%
(i)	Porcentagem (a ou b ou c ou d)/(g)	0.00%	$< 0.01\%$	0.48%	10.44%

O caso (e) não apresentou nenhuma modificação das triplas com adição de um link. O caso (f), ocorreu 245 casos de modificações das triplas com remoção de um link. Porém, isto apenas representou 0,04%, id (h), das modificações.

O caso (g), ocorreu 21.899 casos de modificações das triplas em conjunto de modificação dos links. Isto representou 4.14%, id (h), do total das modificações. Porém

entre as modificações que tem efeito sob os links, casos (e), (f) e (g), é o mais significativa representando um total de 98.89%.

O caso (h), ocorreu 474.253 casos de modificações das triplas sem modificação no link. Portanto, nota-se que este caso é o mais significativo das modificações representado por 95.82%, id (h), das modificações ocorridas.

5 Discussão

Este trabalho visou estudar os efeitos das mudanças em triplas e seus efeitos sob os links. Identificamos a existência de poucos trabalhos na literatura que abordam este assunto, ainda que a manutenção de links tenha um papel importante na manutenção de dados interconectados abertos na Web. Os experimentos propostos neste trabalho obtiveram resultados relevantes que indicam que as modificações têm um papel fundamental na evolução dos dados.

A manutenção dos links causada pelas mudanças do dataset é essencial para a consistência dos dados no modelo RDF. Os resultados revelaram que as mudanças complexas, ou seja, triplas modificadas, que são representadas pelos casos (e), (f), (g) ou (h), são mais significativas que as mudanças simples (adições e remoções). Entre as modificações, verificamos que a mais significativa para a mudança nos links é a representada pelo caso (h), ou seja, uma modificação de um predicado ou objeto da tripla sem a ocorrência da mudança do link relacionado ao sujeito. Dentre as mudanças simples, a operação mais significativa é a adição, em específico o caso (c), isto é, adição de uma nova instância sem a criação de um link.

Os resultados das análises considerando as versões do *Agrovoc* indicaram que na maioria dos casos as modificações complexas têm um impacto maior na evolução dos dados e também tem um maior impacto nas modificações dos links. Isso pode ser observado ao comparar-se as três tabelas (Tabela 1, Tabela 2, Tabela 3) da Seção 4, na qual a modificação representa 98.65% do *Diff* e das mudanças que impactam os links, ou seja, os casos (a), (b), (c), (d), (e), (f), (g), e (h); o caso (h) representou 73.71% do total dos casos enumerados. Igualmente, descobrimos casos de links quebrados - caso (b), que afetam diretamente a qualidade dos datasets, dada a constante evolução dos mesmos.

Como proposta de trabalho futuro, pretendemos explorar casos de mudanças envolvendo a definição de classes das ontologias, e não somente as instâncias. Desenvolveremos algoritmos para lidar com os seguintes casos:

Impacto da remoção de elementos da ontologia em triplas e links.

$Rem(O^j, O^{j+1}) \Rightarrow Add(link)$ ou $Rem(link)$: neste caso, o efeito da remoção ocorrida entre as versões O^j, O^{j+1} serão estudados, analisaremos se a remoção no nível da ontologia tem influência nos links que são formados ou removidos no nível das instâncias.

Impacto da adição de elementos da ontologia em triplas e links.
$Add(O^j, O^{j+1}) \Rightarrow Add(link)$ ou $Rem(link)$: neste caso, o efeito da adição ocorrida entre as versões O^j, O^{j+1} serão estudados, analisaremos se a adição no nível da ontologia tem influência nos links que são formados ou removidos no nível das instâncias.

Ainda como trabalho futuro, pretendemos expandir o alcance da ferramenta de software implementada em razão dos experimentos realizados, tanto horizontalmente - testando com um número maior de datasets de diferentes domínios - quanto verticalmente - aumentando o tamanho dos datasets em análise.

6 Conclusão

O entendimento da evolução de links é essencial para a concepção de mecanismos de manutenção de links na Web Semântica. Este relatório descreveu análises experimentais para entender o comportamento e a correlação entre mudanças ocorridas em triplas que implicam em mudanças nos links para outros datasets. Constatamos quais tipos de mudanças prevalecem no contexto de um dataset do domínio de agricultura que evolui constantemente. Nossos resultados serão relevantes para apoiar a elaboração de uma técnica para lidar com a quebra de links na Web Semântica quando há novas versões de um dataset. Os resultados das análises servirão para se definir e implementar ações corretivas em links afetados pelas mudanças.

Agradecimentos

Este trabalho foi apoiado financeiramente pela FAPESP (processos #2017/02325-5, #2018/05357-8 e #2018/14199-7).

Referências

[1] Berners-Lee, T., Connolly, D.: Delta: an ontology for the distribution of differences between rdf graphs. World Wide Web, http://www. w3. org/DesignIssues/Diff **4**(3), 4–3 (2004)

[2] Galani, T., Papastefanatos, G., Stavrakas, Y.: A language for defining and detecting interrelated complex changes on rdf (s) knowledge bases. In: ICEIS (1). pp. 472–481 (2016)

[3] Galani, T., Stavrakas, Y., Papastefanatos, G., Flouris, G.: Supporting complex changes in rdf (s) knowledge bases. In: DIACRON@ ESWC. pp. 28–33 (2015)

[4] Hogan, A.: Skolemising blank nodes while preserving isomorphism. In: Proceedings of the 24th International Conference on World Wide Web. pp. 430–440. International World Wide Web Conferences Steering Committee (2015)

[5] Käfer, T., Wins, A., Acosta, M.: Modelling and analysing dynamic linked data using RDF and SPARQL. In: Proceedings of the 4th International Workshop on Dataset PROFIling and fEderated Search for Web Data (PROFILES 2017) co-located with The 16th International Semantic Web Conference (ISWC 2017). vol. 1927 (2017)

[6] Meimaris, M., Papastefanatos, G., Pateritsas, C., Galani, T., Stavrakas, Y.: Towards a framework for managing evolving information resources on the data web. In: Proceedings of the 1st International Workshop on Dataset PROFIling fEderated Search for Linked Data co-located with The 16th International Semantic Web Conference (ISWC 2017). vol. 1151 (2014)

[7] Papastefanatos, G., Stavrakas, Y., Galani, T.: Capturing the history and change structure of evolving data. Proceedings of DBKDA 2013 pp. 235–241 (2013)