Uma Abordagem para Caracterização de documentos RDF através de Esquemas Conceituais

Alisson S. Maia[1], **Vagner Pagotti**[1], **Rebeca Schroeder**[1]

[1]Departamento de Ciências da Computação
Universidade do Estado de Santa Catarina (UDESC)
Centro de Ciências Tecnológicas – 89.219-710 – Joinville – SC – Brasil

{alisson.maia11,pagotti}@gmail.com, rebeca.schroeder@udesc.br

Abstract. *A suitable storage model for RDF depends on a set of data characteristics and the knowledgment of its schema. This paper aims to contribute in this context on providing a method to extract conceptual schemas from RDF documents. The goal is to characterize an RDF data structure through an entity-relationship schema and its constructors. The proposed method is evaluated by a case study which demonstrates that the conceptual schemas generated are valid according to the model proposed by a benchmark for RDF.*

Resumo. *Dentre as possibilidades de bancos de dados para o armazenamento de dados RDF, a escolha de um modelo adequado depende de um conjunto de características dos dados e a compreensão de seu esquema. Este trabalho visa contribuir para este contexto através de um método de extração de esquemas conceituais a partir de documentos RDF. O objetivo deste método é caracterizar a estrutura de dados RDF através da produção de um esquema entidade-relacionamento e seus construtores. O método proposto foi avaliado por um estudo de caso que demonstrou que os esquemas conceituais gerados são válidos de acordo com o modelo proposto por um benchmark para RDF.*

1. Introdução

Como um modelo em evidência no contexto da Web Semântica, RDF se tornou alvo de uma série de trabalhos que propõem metodologias para armazenar seus documentos em diferentes tipos de Sistemas de Banco de Dados. Grande parte destes trabalhos apresentam suas propostas de armazenamento RDF para o modelo relacional [Scabora et al. 2017]. Uma vez que dados RDF são descritos no formato de triplas contendo sujeito, predicado e objeto, o mapeamento direto para o relacional corresponde à construção de uma única tabela contendo estes 3 campos. Bancos de dados relacionais que adotam esta estratégia são conhecidos como *triple-stores* [Neumann and Weikum 2010]. Porém, esse método de armazenamento não possui bom desempenho, uma vez que consultas nessa tabela implicam na execução de auto-junções para recuperar triplas relacionadas [Zeng et al. 2013]. Neste caso, fica evidente que compreender a estrutura de dados RDF, e como eles se relacionam, pode favorecer à construção de esquemas de bancos de dados mais apropriados. Embora RDF seja qualificado como um modelo livre de esquema, os autores de [Pham et al. 2015] identificaram que é possível extrair a estrutura de dados RDF para um grande número de *datasets* deste modelo.

Como uma alternativa aos *triple-stores*, alguns trabalhos propõem observar a estrutura dos dados RDF para criar um esquema relacional capaz de agrupar em tabelas

os atributos de uma mesma classe de dados [Ramanujam et al. 2009] [Pham et al. 2015]. Entretanto, a noção das estruturas extraídas são diretamente representadas no modelo relacional, o que pode limitar a representação destes dados através de outros modelos de banco de dados. Neste sentido, este trabalho propõe extrair a estrutura de dados RDF e representá-la através de um esquema conceitual definido através do modelo entidade-relacionamento (ER). Além de servir como base para o projeto de qualquer modelo de banco de dados, a abstração fornecida por esquemas conceituais pode contribuir como uma visão unificada dos conceitos de um domínio relacionado a um conjunto de dados RDF.

A proposta deste trabalho também se diferencia dos trabalhos de [Ramanujam et al. 2009] e [Pham et al. 2015] por extrair algumas métricas que caracterizam a variabilidade da estrutura de *datasets* RDF. Estas métricas dizem respeito às cardinalidades de relacionamentos e atributos do esquema conceitual. A extração destas métricas foi inspirada pelo trabalho de [Duan et al. 2011], em que informações similares foram extraídos diretamente de documentos RDF. Diferente deste trabalho, este artigo apresenta métricas equivalentes mas sobre um esquema conceitual. Acredita-se que tal conhecimento sobre a estruturabilidade dos dados em nível conceitual possa embasar decisões em um futuro projeto de banco de dados, desde a escolha pelo modelo de dados mais apropriado, até o esquema lógico mais adequado.

Este artigo está organizado em mais 3 seções. A Seção 2 apresenta o método de extração de esquemas conceituais a partir de documentos RDF. A seção seguinte apresenta um estudo de caso utilizando a metodologia proposta sobre um gerador de *datasets* de um *benchmark* para RDF. As conclusões deste trabalho são apresentadas pela Seção 4, em conjunto com os trabalhos futuros.

2. Caracterização de Documentos RDF

Dados RDF são representados por triplas definidas por expressões do tipo `sujeito-propriedade-objeto (s, p, o)`. Um documento RDF é constituído por um conjunto de triplas, e pode ser representado por um grafo direcionado, onde os vértices são sujeitos e objetos, e as arestas correspondem às propriedades que os interliga. Um exemplo deste tipo de grafo é apresentado na parte direita da Figura 1. A aresta rotulada com `feature` que conecta os vértices `Produtct1` e `ProductFeature1` representa, por exemplo, a tripla `Produtct1-feature-ProductFeature1`.

Embora RDF corresponda a um modelo livre de esquema, é possível extrair tipos de dados de seus próprios documentos assim como apontado em [Duan et al. 2011]. Em um documento, triplas com a propriedade `http://www.w3.org/1999/02/22-rdf-syntax-ns#type` (`type`) determinam o tipo de seus respectivos sujeitos. A Figura 1 ilustra a extração destes tipos. Por exemplo, a tripla `Produtct2-type-Product` determina que o sujeito `Product2` é do tipo `Product`. Na Figura, os tipos são diretamente representados por entidades do modelo ER. Observe que, a partir da identificação de tipos, é possível identificar como as instâncias de um tipo são estruturadas em termos de atributos e relacionamentos com instâncias de outros tipos. Para tanto, a abordagem proposta por este artigo se baseia neste tipo de análise para sumarizar a estrutura de instâncias de tipos RDF em um esquema conceitual definido no modelo ER.

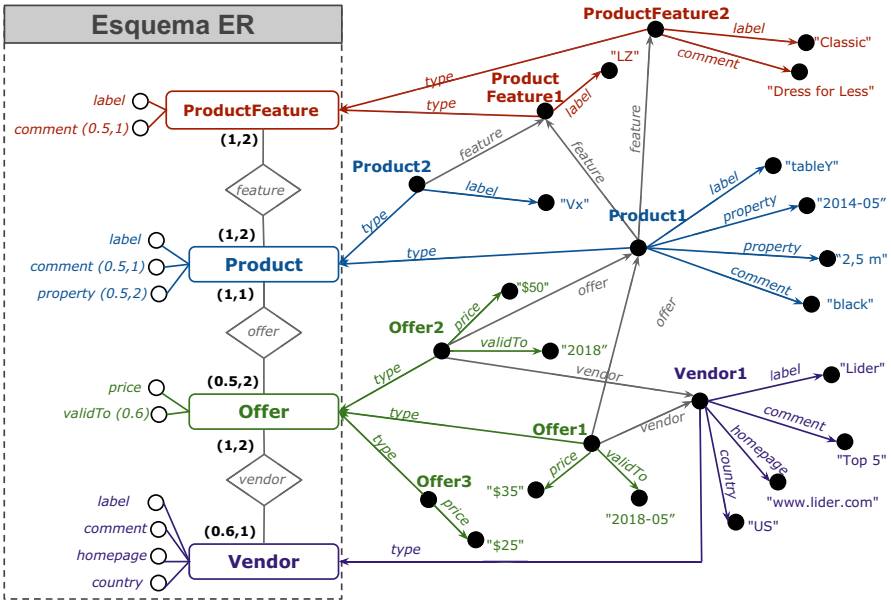

Figura 1. Extração de Esquemas a partir de triplas RDF

Nas seções a seguir são apresentadas as definições que determinam a extração de entidades, atributos e relacionamentos em um esquema ER. Além disto, as cardinalidades de atributos e relacionamentos são apresentadas na sequência.

2.1. Extração de Entidades, Atributos e Relacionamentos

Um esquema conceitual ER S é definido como $S = \{E, R\}$, onde E é o conjunto de entidades e R o conjunto de relacionamentos que representam associações entre duas entidades. Assim como ilustrado pela Figura 1, o conjunto de entidades pode ser extraído de um documento RDF a partir de triplas contendo a propriedade `type`. Logo, a partir de um documento RDF D, o conjunto de entidades é obtido por $E = \{e | \exists(s, \texttt{type}, e) \in D\}$.

No exemplo, E corresponde a $\{\texttt{ProductFeature}, \texttt{Product}, \texttt{Offer}, \texttt{Vendor}\}$. Assim, o conjunto de instâncias de uma entidade $e \in E$ pode ser definido como $I(e) = \{s | \exists(s, \texttt{type}, e) \in D\}$. Por exemplo, $I(\texttt{Product}) = \{Product1, Product2\}$. A partir deste ponto, será utilizado $I(E)$ para representar o conjunto de instâncias de todas as entidades em E. No exemplo, $I(E) = \{\texttt{Product1}, \texttt{Product2}, \texttt{ProductFeature1}, \texttt{ProductFeature2}, \texttt{Offer1}, \texttt{Offer2}, \texttt{Vendor1}\}$.

Os atributos de entidades correspondem a propriedades em D que associam instâncias de E com valores literais. O conjunto de atributos de uma entidade $e \in E$ é definido por $A(e) = \{p | s \in I(e), o \notin I(E), \exists(s, p, o) \in D\}$. Logo, considera-se como valores literais os objetos que não correspondam a instâncias de E. No exemplo, os atributos de `Product` são definidos como $A(\texttt{Product}) = \{\texttt{label}, \texttt{comment}, \texttt{property}\}$. Embora não apresentado no exemplo, considera-se para todas as entidades que um atributo identificador será automaticamente criado, cujo valor conterá a URI de sujeitos que correspondem a elementos de $I(E)$. Além disto, atributos obrigatórios e opcionais serão identificados pela extração de suas respectivas cardinalidades a ser apresentado na

próxima seção.

Os relacionamentos são extraídos a partir das propriedades que associam duas instâncias diferentes de E. Logo, o conjunto de relacionamentos de S é definido por $R = \{r | s \in I(E), o \in I(E), \exists(s, r, o) \in D\}$. De acordo com o exemplo, $R = \{\texttt{feature}, \texttt{offer}, \texttt{vendor}\}$. Dado que D é um grafo direcionado, denomina-se $\texttt{out}(r)$ a entidade que corresponde à entidade origem da direção apontada pelo relacionamento, assim como $\texttt{in}(r)$ a entidade destino. No exemplo, $\texttt{out}(\texttt{feature})$ corresponde à entidade `Product` e $\texttt{in}(\texttt{feature})$ à entidade `ProductFeature`. Uma vez que os relacionamentos são extraídos diretamente pelas associações estabelecidas entre sujeitos e objetos de triplas RDF, não é possível a extração de relacionamentos n-ários e de atributos para relacionamentos. Entretanto, considera-se que ambas construções de um modelo ER possam ser respectivamente representadas por um conjunto de relacionamentos binários e atributos de entidades relacionadas. Ademais, o método de extração das cardinalidades mínimas e máximas dos relacionamentos é apresentado pela próxima seção.

2.2. Extração de Cardinalidades

Como apresentado pela Seção 2.1, tanto os atributos quanto os relacionamentos são extraídos a partir de propriedades das triplas de D. Da mesma forma, conhecer a frequência com que estas propriedades ocorrem para instâncias de entidade determina a cardinalidade de atributos e relacionamentos associados.

Para atributos de entidade, a cardinalidade mínima define a quantidade mínima de valores associados a cada instância de uma entidade, qualificando atributos opcionais ou obrigatórios. A cardinalidade mínima para um atributo a de uma entidade e é definida por:

$$min\_card(a) = \frac{| \{s | s \in I(e),\ a \in A(e),\ \exists(s, a, o) \in D\} |}{| I(e) |} \quad (1)$$

Observe que a Equação 4 não tem por objetivo contabilizar a quantidade de ocorrências de uma dada propriedade sobre todas as instâncias de e, e sim a quantidade de instâncias que possuem tal propriedade associada. Assim sendo, dadas as duas instâncias de `Product` do exemplo, o atributo `label` é qualificado como obrigatório, isto é $\texttt{min\_card(label)}=1$, visto que as duas instâncias de `Product` contêm esta propriedade. Entretanto, para o atributo `comment` a cardinalidade mínima obtida é 0.5, uma vez que apenas `Product1` contém esta propriedade. Logo, `comment` corresponde a um atributo opcional. Desta forma, a cardinalidade mínima de um atributo pode assumir valores entre 0 e 1 (inclusive). Por se tratar de um modelo semi-estruturado, valores de cardinalidade menores que 1 indicam não somente que o atributo é opcional, mas também a proporção de instâncias de entidade que apresentam valores para esta propriedade.

A cardinalidade máxima de um atributo é extraída com base na ocorrência máxima encontrada em uma das instâncias de sua respectiva entidade. Para tanto, dado o total de instâncias de uma entidade e, a cardinalidade máxima de um atributo a desta entidade é definida por uma instância e_i que maximiza a seguinte expressão:

$$max\_card(a) = max(| \{a | e_i \in I(e),\ a \in A(e),\ \exists(e_i, a, o) \in D\} |) \quad (2)$$

De acordo com o exemplo da Figura 1, a cardinalidade máxima do atributo `property` em `Product` é igual a 2. Quanto aos demais atributos, a ocorrência máxima

é igual a 1. Para efeito de simplificação do exemplo, atributos que não apresentam cardinalidades mínimas e máximas anotadas correspondem a (1,1), isto é, atributos obrigatórios e mono-valorados.

De forma análoga aos atributos, as cardinalidades dos relacionamentos são obtidas pela avaliação das propriedades associadas. Entretanto, neste caso, as cardinalidades são dadas para cada uma das entidades que participam de um relacionamento. Logo, dado um relacionamento r definido entre duas entidades, a cardinalidade mínima de cada uma das entidades participantes é definida por:

$$min\_card(e, r) = \frac{\mid \{e | e \in I(E),\ r \in R,\ (\exists(e, r, o) \in D\ ou\ \exists(s, r, e) \in D)\} \mid}{\mid I(e) \mid} \quad (3)$$

Em virtude da direção das arestas em D, a Equação 6 considera que as arestas relacionadas a r são dadas em D em apenas uma direção. Desta forma, se $\texttt{out}(\texttt{r}) = e$ haverá uma tripla $(e, r, o) \in D$ para uma das instâncias de e. Caso contrário, se $\texttt{in}(\texttt{r}) = e$, haverá uma tripla $(s, r, e) \in D$. Como exemplo, a cardinalidade mínima da entidade `Offer` no relacionamento `vendor` é igual a 0.6, uma vez que das 3 instâncias de `Offer`, apenas 2 estão relacionadas por `vendor`. Entretanto, a cardinalidade mínima de `Vendor` no relacionamento `vendor` é igual a 1 em virtude de que a única instância desta entidade está associada pelo relacionamento.

A cardinalidade máxima de uma entidade e em um relacionamento r é extraída com base na ocorrência máxima encontrada em uma das instâncias de suas entidades. Para tanto, dado o total de instâncias de uma entidade e, sua cardinalidade máxima em r é definida por uma instância e_i que maximiza a seguinte expressão:

$$max\_card(e, r) = max(\mid \{a | e_i \in I(e),\ (\exists(e_i, r, o) \in D\ ou\ \exists(s, r, e_i) \in D)\} \mid) \quad (4)$$

No exemplo da Figura 1, $\texttt{max\_card}(\texttt{Product}, \texttt{offer}) = 2$ cujo valor máximo é determinado pela instância `Product1`. Já $\texttt{max\_card}(\texttt{Offer}, \texttt{offer}) = 1$, visto que todas as instâncias de `Offer` estão associadas a no máximo 1 instância de `Product`.

A extração de entidades, atributos e relacionamentos pode ser derivada de abordagens que fazem o mapeamento do modelo RDF para o relacional. Entretanto, em virtude da fraca abstração do modelo relacional, a noção de cardinalidades mínimas e máximas não pode ser identificada. Considera-se que esta noção é fundamental para, por exemplo, evitar a geração de tabelas com muitos atributos opcionais.

3. Estudo de Caso

Um protótipo que implementa o método de extração proposto foi desenvolvido para avaliação dos esquemas produzidos. Para a estudo, foi utilizado o gerador de documentos RDF fornecido pelo *Berlin SPARQL Benchmark* (BSBM)[Bizer and Schultz 2009]. O BSBM é baseado em um caso de uso de um sistema de *e-commerce*, onde uma lista de produtos é oferecida por vendedores. Parte do esquema de dados do BSBM foi utilizado no exemplo da Figura 1. Para a geração das bases, o *benchmark* utiliza um fator de escala baseado no número de produtos a gerar. Neste estudo aplicou-se o fator de 100 produtos, produzindo um documento com 50 mil triplas RDF.

Um esquema ER é apresentado pela própria documentação do BSBM. Para tanto, comparou-se o esquema ER do BSBM com o produzido pelo protótipo do método aqui proposto. Observou-se que todas as entidades, atributos e relacionamentos foram extraídos adequadamente. No entanto, com relação às cardinalidades houve pequenas divergências relacionadas às cardinalidades mínimas de 3 entidades em seus relacionamentos. Neste caso, o documento RDF foi verificado e constatou-se que as cardinalidades não correspondiam às indicadas na documentação do BSBM, e sim às geradas pelo protótipo desenvolvido.

4. Conclusões

Este artigo propõe um método para a extração de esquemas conceituais ER a partir de documentos RDF. A solução apresentada difere de trabalhos relacionados por representar a estrutura de dados através de um modelo mais abstrato, se comparado ao modelo relacional utilizado por estes trabalhos. Os esquemas produzidos pelo método proposto podem ser utilizados para apoiar decisões do projeto de um BD para armazenamento de dados RDF. Por exemplo, o conhecimento da cardinalidade mínima de atributos pode evitar a criação de tabelas com excesso de colunas opcionais, caso um BD relacional venha a ser escolhido. Sobretudo, os esquemas produzidos são capazes de expressar o grau de estruturabilidade de dados RDF através da opcionalidade de relacionamentos e atributos indicado por suas respectivas cardinalidades.

Um estudo de caso foi realizado em que se verificou que os esquemas produzidos por este trabalho correspondem aos esquemas de um *benchmark* para RDF. Entretanto, experimentos com *datasets* reais e maiores se fazem necessários para a validação do método sobre fontes com um grau de estruturabilidade provavelmente menor. Além disto, considera-se como trabalho futuro a identificação de variações na estrutura dos dados em decorrência à evolução de esquemas RDF.

Referências

Bizer, C. and Schultz, A. (2009). The Berlin SPARQL Benchmark. *Int. J. Semantic Web Inf. Syst.*, 5(2):1–24.

Duan, S., Kementsietsidis, A., S., K., and U., O. (2011). Apples and oranges: A Comparison of RDF Benchmarks and Real RDF Datasets. In *SIGMOD*, pages 145–156.

Neumann, T. and Weikum, G. (2010). The RDF-3X Engine for Scalable Management of RDF Data. *The VLDB Journal*, 19(1):91–113.

Pham, M.-D., Passing, L., Erling, O., and Boncz, P. (2015). Deriving an Emergent Relational Schema from RDF Data. In *WWW*, pages 864–874.

Ramanujam, S., Gupta, A., Khan, L., Thuraisingham, B., and Seida, S. (2009). R2D: Extracting Relational Structure from RDF Stores. In *International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 361–366.

Scabora, L. C., Oliveira, P. H., dos Santos Kaster, D., Traina, A. J. M., and Traina, C. (2017). Relational graph data management on the edge: Grouping vertices' neighborhood with Edge-k. In *Simpósio Brasileiro de Banco de Dados*, pages 124–135.

Zeng, K., Yang, J., Wang, H., Shao, B., and Wang, Z. (2013). A Distributed Graph Engine for Web Scale RDF Data. *Proceedings of the VLDB Endowment*, 6(4):265–276.