OntoGenesis: Uma Arquitetura para Enriquecimento Semântico de Web Data Services

Bruno C. N. Oliveira
bruno.cno@posgrad.ufsc.br
Programa de Pós-graduação em Ciência da Computação
Universidade Federal de Santa Catarina
Florianópolis, Santa Catarina, Brasil 88040-900

Frank Siqueira
frank.siqueira@ufsc.br
Departamento de Informática e Estatística
Universidade Federal de Santa Catarina
Florianópolis, Santa Catarina, Brasil 88040-900

ABSTRACT

In recent years, many approaches and tools have emerged to assist in the semantic enrichment of Web Services. Many researchers have been directing efforts in enriching service descriptions. However, the automatic enrichment of data representations provided by Web Services has been of little concern in the literature. This work aims to present OntoGenesis, an architecture capable of constructing and evolving domain ontologies for data services and enriching their data with semantic concepts from such ontologies. The proposed solution also takes into account external semantic sources to enhance the reuse of well-known concepts by means of ontology property matching techniques. Preliminary results show the applicability of our approach under a scenario within real-world datasets.

KEYWORDS

Data Services, Semantic Web, Semantic Enrichment, Ontology Construction, Property Matching

1 INTRODUÇÃO

O número de Web *Services*, espalhados em redes públicas e privadas, que se destinam a fornecer dados armazenados em alguma fonte de dados tem aumentado consideravelmente. Tais Web *Services*, também chamados de Web *data services* (ou apenas serviços de dados), têm como objetivo fornecer interfaces Web de acesso e manipulação a fontes de dados [6]. O rápido crescimento de tais serviços e fontes de dados disponibilizadas hoje na Web resultou em uma disseminação de dados representados em formatos heterogêneos e difíceis de serem processados e reutilizados.

Em virtude disso, as tecnologias da Web Semântica [3] surgem como um meio de prover significado aos dados disponíveis na Web, promovendo a sua compreensão não apenas por humanos, mas também por máquinas. Adicionalmente, os serviços de dados podem empregar tais tecnologias de modo a prover dados em um formato mais sofisticado processado por máquinas, e facilitar a reutilização e a integração com aplicações Web mais complexas. Diversos pesquisadores apontam os benefícios de se empregar Web *Services* Semânticos, tanto na interoperabilidade dos dados [17, 20] quanto no auxílio para automatizar processos de descoberta, seleção e composição de serviços, dentre outros [15].

A implementação e a adoção de serviços de dados semânticos (SDS), todavia, é limitada principalmente pelo formato dos dados armazenados, requerendo que estes estejam descritos semanticamente. Em geral, os dados estão armazenados sintaticamente, isto é, apenas a estrutura dos dados é especificada, mas não a semântica. Além disso, vários desafios contribuem para essa falta de adoção. Alguns problemas comuns incluem o tempo e o esforço exigidos na construção de ontologias de domínio e a anotação semântica dos serviços de dados. Essas são tarefas complexas que exigem alto conhecimento do domínio em questão. Além disso, ainda existe o hiato entre construir um serviço com suporte semântico e a possibilidade de explorar um repositório de ontologias abrangendo diversos domínios. Na prática, não é realista assumir que os dados fornecidos pelos serviços serão sempre definidos por uma ontologia universal [10]. Isso traz um novo desafio para o desenvolvimento e integração de SDS, devido à existência de diferentes ontologias que descrevem a mesma entidade do mundo real.

Visto que a construção de uma ontologia para uma fonte de dados é uma tarefa difícil e onerosa por natureza, diversas ferramentas surgiram para apoiar os desenvolvedores e especialistas de domínio no processo de construção de ontologias [7, 16, 19]. Entretanto, tais ferramentas muitas vezes exigem a disponibilidade de um *dump* dos dados para gerar uma ontologia de domínio. Esta limitação dificulta a adoção de tais ferramentas em um ambiente SOA (*Service Oriented Architecture*), onde os dados são disponibilizados através de uma interface de serviço. Já as técnicas de *matching* de ontologia [9] extensional tentam resolver o problema da heterogeneidade de ontologias usando as instâncias de dados para inferir equivalências.

Por outro lado, os serviços de dados estão suscetíveis a mudanças e as ontologias que descrevem os dados devem necessariamente evoluir em paralelo, caso contrário elas se tornam inconsistentes. Além disso, para realizar o *matching* extensional, os *matchers* de ontologia geralmente assumem que as ontologias de entrada já foram criadas e associadas a uma grande quantidade de instâncias de dados. Isso se torna um desafio quando esses dados são parcialmente fornecidos por interfaces de serviço e quando não há uma relação direta entre os dados descritos pelas ontologias. Yao et al. [23] introduzem um mecanismo para construir uma ontologia unificada a partir de um conjunto de documentos JSON fornecidos por serviços de dados. No entanto, a ontologia criada não emprega conceitos semânticos definidos em fontes externas com o objetivo de reutilizar conceitos existentes e minimizar problemas de interoperabilidade. Além disso, em estudos anteriores, pesquisadores direcionaram esforços em abordagens para enriquecer semanticamente Web *Services* [22], sendo que a maioria deles se concentra no enriquecimento da descrição dos serviços [5, 12, 13]. Em contrapartida, poucas propostas abordam o enriquecimento dos dados

In:XVII Workshop de Teses de Dissertações (WTD 2017), Gramado, Brasil. Anais do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web: Workshops e Pôsteres. Porto Alegre: Sociedade Brasileira de Computação, 2017.

ISBN 978-85-7669-380-2.

fornecidos por serviços. Salvadori et al. [20], por exemplo, propõem um método para enriquecer representações de *data-based microservices* utilizando links `owl:sameAs` e `rdfs:seeAlso`. Além disso, os autores também propõem um *framework* que visa identificar alinhamentos entre ontologias heterogêneas. Um inconveniente desta abordagem é que os *microservices* já devem empregar uma ontologia de domínio, bem como fornecer dados semânticos.

Em vista do exposto, foi formulada a seguinte hipótese: serviços de dados podem ser enriquecidos automaticamente, fornecendo representações descritas semanticamente a partir da reutilização de diferentes recursos já existentes. O objetivo deste trabalho, portanto, visa à concepção, desenvolvimento e avaliação de uma arquitetura que forneça suporte i) à construção e evolução de ontologias de domínio; e ii) ao enriquecimento automático de serviços de dados com os conceitos semânticos definidos em tais ontologias.

Como contribuição, a abordagem proposta possibilitará uma maneira de construir e evoluir progressivamente ontologias de domínio a partir de representações sintáticas fornecidas por serviços de dados, além de reutilizar conceitos já existentes por meio de técnicas de *matching* de ontologias. Espera-se, pois, abstrair dos desenvolvedores de *software* o processo oneroso de enriquecer semanticamente serviços conectados a fontes que armazenam dados sem semântica. Ademais, a proposta avança o estado da arte no sentido de prover uma nova abordagem que permita migrar serviços de dados definidos sintaticamente para *data services* semânticos. Desse modo, tarefas mais sofisticadas podem ser executadas, dando ensejo à reutilização e integração de dados entre diversas aplicações.

2 FUNDAMENTAÇÃO TEÓRICA

Para tornar os dados publicamente disponíveis na Web, as fontes de dados geralmente são disponibilizadas por meio de *Web Services*, os quais fornecem uma interface Web para lidar com dados. Tais serviços são chamados de Web *data services* (neste trabalho consideramos serviços de dados como sinônimo). Bianchini et al. [4] definem um serviço de dados como uma operação, método ou consulta para acessar dados de uma determinada fonte de dados. Esses serviços são modelados como um conjunto de: i) entradas, que consistem em parâmetros necessários para invocar o serviço; e ii) saídas, representando os dados que são acessados através do serviço. A representação de saída pode ser vista como um *snaphshot* do estado de um recurso em um determinado momento, disponível em diferentes formatos, como XML, JSON, HTML, etc. A principal característica dos serviços de dados é fornecer uma abstração de acesso às fontes de dados, atuando como provedores de dados.

Um fator crucial que dificulta a integração de tais serviços ocorre no nível conceitual, uma vez que são frequentemente empregadas terminologias distintas (por exemplo, diferentes nomes de atributos), mesmo se tratando de informações sobre o mesmo conceito do mundo real. Para superar tal inconveniente, as tecnologias da Web Semântica [3], tais como ontologias, RDF e *Linked Data*, podem ser aplicadas aos serviços de dados, resultando em serviços de dados semânticos (SDS). Tais serviços surgem como uma forma de reduzir o extenso esforço necessário para manipular *Web Services* tradicionais (sintáticos) e facilitar processos como descoberta e composição de serviços [15]. Desse modo, SDS utilizam ontologias com a finalidade de descrever os seus dados e fornecer aos consumidores informações semanticamente enriquecidas.

As ontologias desempenham um papel fundamental na Web Semântica, especialmente no âmbito de Web *Services*. Em geral, engenheiros de ontologia e especialistas do domínio desenvolvem manualmente ontologias para fornecer um modelo específico de domínio adequado para descrever a semântica de um serviço. O maior esforço envolvido durante o processo de enriquecimento semântico de serviços está, portanto, na construção de ontologias, bem como na sua adaptação e evolução, de acordo com as mudanças demandadas. *Ontology Learning* (OL) [14] é uma área de pesquisa que busca automatizar e, por conseguinte, facilitar a construção de ontologias. Desta forma, elementos ontológicos, como conceitos e suas relações, são extraídos de diferentes recursos de forma (semi-)automática, podendo utilizar os recursos fornecidos por Web Services semânticos como fonte para um processo de OL [1].

Embora OL ofereça mecanismos para automatizar o processo de construção de ontologias, é essencial que os conceitos das diversas ontologias existentes sejam reutilizados. Além de ampliar as possibilidades de integração, isso permite que operações de *reasoning* sejam executadas por agentes de *software*. Neste sentido, as técnicas de *matching* de ontologia surgem para resolver questões de heterogeneidade de ontologias, identificando as correspondências – geralmente expressas por relações de equivalências – entre diferentes ontologias. Euzenat e Shvaiko [9] identificam quatro técnicas básicas para *matching* de ontologias. A técnica baseada em nomes considera apenas o nome dos elementos da ontologia (como os rótulos das propriedades e classes). A técnica estrutural considera a estrutura dos elementos ontológicos (e.g., as relações de subclasse). As técnicas baseadas em semântica, por outro lado, geralmente executam *reasoning* de modo a inferir as equivalências entre diferentes ontologias. Finalmente, as técnicas extensionais fazem uso das instâncias das ontologias para encontrar indivíduos semelhantes e, assim, identificar classes e propriedades correspondentes.

Visto que os serviços de dados fornecem informações de fontes de dados, as técnicas de *matching* extensional podem ser adaptadas de tal forma que os dados providos pelos serviços possam ser adicionados como instâncias da ontologia. Assim, além da construção de ontologias de domínio para serviços de dados, este trabalho também compreende técnicas de *matching* extensional para identificar alinhamentos entre as ontologias criadas para os serviços de dados e ontologias externas já existentes, focando mais precisamente no alinhamento de suas propriedades. Tais alinhamentos são denotados pelo axioma `owl:equivalentProperty` [2].

3 TRABALHOS RELACIONADOS

Os trabalhos relacionados foram divididos em duas categorias. A primeira discute trabalhos que focam no *matching* e construção de ontologias a partir de uma fonte de dados. A segunda discute os trabalhos que abordam o enriquecimento semântico de serviços.

As técnicas de *matching* de ontologia [9] apresentam desafios específicos no cenário de enriquecimento semântico de serviços. A técnica que mais se assemelha ao problema que atacamos neste trabalho é o *matching* extensional, a qual pode ser adaptada para o enriquecimento de serviço, conforme visto em [20]. Tais técnicas, no entanto, buscam identificar correspondências entre duas ontologias a partir do compartilhamento dos seus indivíduos [8, 21]. O

AROMA [8], por exemplo, produz alinhamentos entre ontologias quando os seus indivíduos são equivalentes. Caso os indivíduos não compartilhem pelo menos duas propriedades, o alinhamento não é bem sucedido. Antagonicamente, nossa abordagem além de construir dinamicamente ontologias de domínio, permite que correspondências sejam feitas no nível de propriedades, i.e., realiza o *matching* das ontologias com base nos valores de suas propriedades.

Com relação à construção de ontologias, diversas ferramentas [7, 16, 19] foram projetadas com o objetivo de apoiar os usuários e especialistas de domínio neste processo. Não obstante, todas elas sofrem de alguma deficiência. Primeiro, a maioria delas depende de modelos de ontologias muito específicos ou proprietários, o que dificulta a sua ampla aplicabilidade. Além disso, tais ferramentas apenas auxiliam os usuários a criar ontologias, não sendo, portanto, abordagens totalmente automáticas. Finalmente, os métodos tradicionais de construção de ontologias, em geral, exigem como entrada um enorme conjunto de dados não estruturados ou de páginas Web [16], diferentemente da nossa abordagem, na qual os dados são fornecidos sob demanda pelos serviços.

Yao et al. [23] apresentam um *framework* cujo objetivo é gerar uma ontologia unificada a partir de um conjunto de documentos da Web em formato JSON. Para tanto, os elementos do JSON são, inicialmente, convertidos em triplas RDF similares ao Turtle. Em seguida, é feito um mapeamento semântico para construir ontologias e instâncias baseadas nos metadados dos documentos. Por fim, o *framework* contempla uma fase de *ontology merging*, resultando em uma ontologia unificada. Embora demonstre ser um estudo relevante relacionado à construção de ontologia com base em documentos Web, os autores não abordam o enriquecimento semântico automático dos serviços que provêm o JSON. Além disso, o *framework* só aceita JSON e produz uma única ontologia como saída.

Muitos pesquisadores têm envidado esforços no desenvolvimento de abordagens automáticas/semi-automáticas para enriquecer semanticamente Web *Services* [22]. Estes trabalhos podem ser divididos em dois subgrupos: i) os que tratam do enriquecimento da descrição dos serviços (isto é, adicionam informações semânticas para descrever as suas interfaces); e ii) os que focam no enriquecimento das representações fornecidas pelo serviço, conectando-as com recursos semânticos. Pouca atenção, no entanto, é dada ao último grupo. A maioria das propostas encontradas na literatura tem como objetivo aprimorar as descrições do serviço com semântica, como SDWS [5], SWS Editor [12] e ASSARS [13].

O método proposto por Zhang et al. [24] busca unir as URIs existentes no DBpedia com parâmetros de serviços SOAP. Inicialmente, a abordagem analisa descrições WSDL com o intuito de identificar seus elementos: interfaces, operações e parâmetros. Cada parâmetro então é refinado a fim de regularizar o seu nome, visto que estes muitas vezes são definidos de forma irregular (e.g., com abreviações e sem espaçamento entre as palavras). Por fim, consulta-se a ontologia do DBpedia em busca de conceitos similares aos dos parâmetros já refinados, além de recuperar também as instâncias existentes no DBpedia que possuam correspondência com algum parâmetro do serviço. Para tanto, a abordagem utiliza técnicas de *matching* de ontologias para obter o conceito com maior similaridade semântica e utilizá-lo para anotar os parâmetros do serviço. Apesar de os autores apresentarem um método de anotação semântica para os parâmetros de entrada e saída de um serviço, as suas entidades de retorno não são enriquecidas de modo a prover *Linked Data*.

Tabela 1: Comparativo dos trabalhos relacionados.

Técnicas	[8] [21]	[7] [16] [19]	[23]	[5] [12] [13]	[24]	[18]	[20]
Matching de Ontologia	✓		✓		✓		✓
Construção de Ontologia		✓	✓				
Enriq. da Representação			✓			✓	✓
Enriq. da Descrição				✓	✓		
100% Automática	✓				✓		✓

No âmbito de Web *data services*, Quarteroni et al. [18] propõem um processo semi-automático para registrar serviços, o qual explora bases de conhecimento existentes, bem como técnicas de processamento de texto, para anotação semântica e integração de serviços de dados. Salvadori et al. [20] propõem um método de composição que explora a intersecção de dados observada nas descrições de *data-based microservices* com o intuito de conectar recursos semânticos. As representações fornecidas pelos *microservices* são, então, enriquecidas por meio de links `owl:sameAs` e `rdfs:seeAlso`. Adicionalmente, os autores propõem um *framework* chamado Alignator, o qual adota técnicas de *matching* de ontologias para identificar alinhamentos entre ontologias que descrevem diferentes *microservices*. Esta abordagem, porém, considera apenas serviços que i) já empregam uma ontologia de domínio previamente definida e ii) já fornecem representações semânticas. Portanto, representações sintáticas não são suportadas pelo *framework*.

A Tabela 1 compara os principais trabalhos relacionados de acordo com cinco critérios: se o trabalho aborda *matching* de ontologias; se o trabalho adota técnicas de construção de ontologias; se a abordagem enriquece semanticamente as descrições de serviços, ou as suas representações; e se a abordagem é totalmente automática. É possível observar que nenhum trabalho atende simultaneamente a todos os aspectos levantados, além de a construção de ontologias ser feita apenas de maneira supervisionada (semi-automática).

4 PROPOSTA

Esta seção apresenta o método de pesquisa deste trabalho, a arquitetura proposta do OntoGenesis e os resultados preliminares.

4.1 Metodologia

O método de desenvolvimento do trabalho está dividido em 4 etapas:

Etapa 1: Revisão da Literatura. Inicialmente, a partir de uma colaboração em uma Revisão Sistemática da Literatura (RSL) – a qual adotou a metodologia proposta por Kitchenham [11] – foi identificado como a semântica é considerada no contexto de composição, seleção, descoberta e descrição de serviços. Tal RSL ajudou a entender a importância da semântica no contexto destes processos e a identificar nos trabalhos selecionados como a semântica é adotada no âmbito de Web *Services*. As conclusões oriundas dessa RSL deram subsídio a uma nova revisão bibliográfica dedicada a buscar na literatura abordagens e técnicas adicionais tratando exclusivamente de enriquecimento semântico automático de serviços de dados. Esta etapa, pois, teve como objetivo analisar o estado da arte de forma a garantir a originalidade das técnicas propostas.

Etapa 2: Concepção da Proposta. Nesta etapa buscou-se definir um mecanismo para enriquecimento semântico de serviços de dados de forma automática e em tempo de execução. Como resultado, foi concebida uma arquitetura, chamada OntoGenesis (seção 4.2), a qual atendeu alguns requisitos fundamentais visando cumprir o objetivo proposto. Primeiro, foram contempladas técnicas alinhadas às encontradas na literatura, como *matching* de ontologias e OL, aspirando à construção e à evolução de ontologias de domínio para serviços de dados. Segundo, foi essencial definir um mecanismo para que serviços (que originalmente proveem dados sintáticos) sejam capazes de servir dados com semântica, mais precisamente, *Linked Data*, de modo a abstrair do desenvolvedor grande parte do esforço necessário para construção de SDS. Por fim, o OntoGenesis incorpora no serviço uma interface de acesso à ontologia criada, para que os clientes e sistemas externos possam consumi-la.

Etapa 3: Implementação. O objetivo desta etapa foi implementar a arquitetura do OntoGenesis definida na etapa 2. Com o intuito de avaliar a aplicabilidade da proposta em curto prazo, a implementação foi baseada no processo de prototipação. Portanto, as funcionalidades essenciais da arquitetura (e.g., construção de ontologias de domínio para serviços de dados e mecanismo para realizar *matching* e obter correspondências com outras ontologias externas) foram desenvolvidas e testadas para se obter resultados preliminares (seção 4.3). Após a análise dos resultados, prevê-se a evolução do protótipo contemplando todos os componentes da arquitetura.

Etapa 4: Execução dos Experimentos e Avaliação dos Resultados. Primeiramente foram coletados dados reais para execução de novos experimentos. Em seguida, a proposta foi avaliada observando-se métricas como precisão, cobertura e F-Measure. Por fim, deve-se corroborar a hipótese elicitada neste trabalho e comparar os resultados obtidos com outras abordagens encontradas na literatura, utilizando os mesmos dados e ambientes de execução.

4.2 Arquitetura do OntoGenesis

A Figura 1 apresenta o funcionamento geral da abordagem proposta, juntamente com a arquitetura do OntoGenesis e seus principais componentes, a saber: OntoGenesis *Engine*, OntoGenesis API e *Semantic Adapter*. O primeiro é responsável pela construção de ontologias de domínio e pela produção de mapeamentos semânticos (descritos em seguida). O segundo componente se refere a uma API Web a qual provê uma interface de acesso às funcionalidades providas pelo *Engine*. Por fim, o *Semantic Adapter* trata-se de uma biblioteca para auxiliar serviços de dados no provimento de representações enriquecidas semanticamente por meio da API.

Um mapeamento semântico pode ser definido como uma 3-tupla $\mathcal{MS} = \{a, p, t\}$, onde a é o atributo de uma representação; p é a propriedade representada na ontologia; e t é o tipo da propriedade (*datatype* ou *object property*). Como exemplo, suponha que seja criada uma *datatype property* p, onde p = `"http://service1-ontology#name"`, para o atributo a = `"name"` de uma dada representação. O mapeamento semântico gerado deve ser $\mathcal{MS}$ = {`"name"`, `"http://service1-ontology#name"`, `"Datatype property"`}. Os mapeamentos semânticos são úteis para gerar representações semânticas fornecidas pelos serviço de dados. Desta forma, de acordo com o conjunto de mapeamentos semânticos produzidos pelo OntoGenesis, a representação sintática é convertida automaticamente em

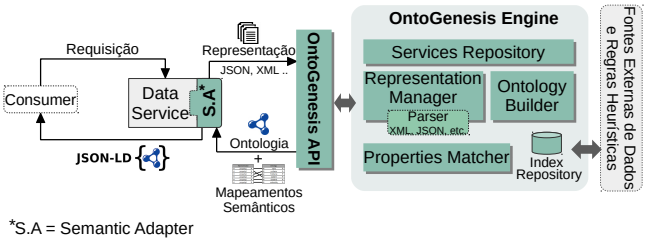

Figura 1: Visão geral da arquitetura do OntoGenesis.

JSON-LD (JSON *Linked Data*) por um adaptador semântico (*Semantic Adapter*), que por sua vez é devolvido ao consumidor. O *Semantic Adapter* é um componente, conectado a um serviço de dados, que intercepta as respostas e retorna representações JSON-LD aos consumidores. Também pode ser visto como um conector responsável pela comunicação entre o serviço de dados e a API do OntoGenesis.

OntoGenesis Engine. Conforme ilustrado na Figura 1, o *Engine* é composto por outros sub-componentes, descritos a seguir.

O *Services Repository* gerencia informações sobre os serviços de dados registrados, como o nome do serviço, o seu endereço (URI) e os recursos semânticos criados pelo OntoGenesis (ou seja, a ontologia do domínio e os mapeamentos semânticos). Tais informações são utilizadas pelos demais componentes do OntoGenesis.

O *Representation Manager* visa a extrair os elementos de uma representação fornecida pelo serviço, como os atributos e seus valores, úteis para o processo de construção da ontologia. Para este fim, ele fornece uma abstração comum para qualquer formato de dados, de modo que *parsers* específicos podem ser encapsulados permitindo que o OntoGenesis *Engine* lide com diferentes formatos de dados, como JSON, XML, CSV, HTML, entre outros.

O *Ontology Builder* analisa os elementos sintáticos extraídos pelo *Representation Manager* para construir uma ontologia de domínio para o serviço registrado. Se uma ontologia de domínio já foi construída a partir de uma representação anterior enviada pelo serviço, o *Ontology Builder* atualiza a ontologia do domínio com os novos elementos identificados. Portanto, a ontologia do serviço evolui à medida que novas representações são fornecidas ao OntoGenesis. A Figura 2 ilustra uma amostra de uma ontologia (em Turtle) (b) com base em uma dada representação em JSON (a). Os valores contidos no JSON representam dados reais – publicados pela Secretaria da Segurança Pública do estado de São Paulo (SSP/SP) – de um Boletim de Ocorrência (BO) com informações sobre uma pessoa envolvida (vítima, testemunha ou autor de um crime).

Embora a ontologia produzida pelo *Ontology Builder* forneça conceitos semânticos relacionados aos dados providos pelo serviço, tais conceitos só são conhecidos pelo serviço de dados. Com o objetivo de permitir uma integração mais rica com outros aplicativos ou serviços existentes no âmbito da Web Semântica, é essencial que a ontologia construída reutilize (ou se alinhe a) conceitos definidos por ontologias/vocabulários abertos e já conhecidos.

O *Index Repository* é um componente que armazena em uma base chave-valor dados oriundos de fontes externas, bem como os dados produzidos pelos serviços. O índice é representado pelas propriedades de entidades contidas em fontes externas e de entidades gerenciadas pelos serviços de dados. Já os valores do índice são representados pelo conjunto de valores literais associados a

```
{ "BoletimOcorrencia": {
      "idBO": "2015-10004-794",
      "local": "Estação do Metrô ...",
      "perssoaEnvolvida": {
          "nome": "CARLOS ALBERTO DOS SANTOS",
          "rg": "015***18",
          "dataNascimento": "12-21-1966",
          "nacionalidade": "Brazilian",
          "localNascimento": "Sao Paulo-SP",
          "genero" : "Male",  ...
} } }
```

(a)

```
@prefix :     <http://exemplo-servico/ontologia#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
<http://exemplo-servico/ontologia> a owl:Ontology .
:BoletimOcorrencia a owl:Class .
:PessoaEnvolvida    a owl:Class .
:temPessoaEnvolvida a owl:ObjectProperty; rdfs:domain  :BoletimOcorrencia;
                                          rdfs:range   :PessoaEnvolvida .
:idBO a owl:DatatypeProperty ;            rdfs:domain  :BoletimOcorrencia;
                                          rdfs:range   xsd:string .
:nome a owl:DatatypeProperty ;            rdfs:domain  :PessoaEnvolvida;
                                          rdfs:range   xsd:string .
:rg a owl:DatatypeProperty ;              rdfs:domain  :ParteEnvolvida;
                                          rdfs:range   xsd:string .
:dataNascimento a owl:DatatypeProperty ;  rdfs:domain  :ParteEnvolvida;
                                          rdfs:range   xsd:date .  # ...
```

(b)

Figura 2: Amostra de: (a) Representação de um BO em JSON; e (b) Ontologia construída a partir da representação.

cada propriedade. Regras Heurísticas também podem ser empregadas como fontes de informação para alinhar as propriedades. Tais regras podem ser descritas como expressões regulares para uma determinada propriedade p. Um simples exemplo de uma regra $\mathcal{R}$ é `dbo:date` $\rightarrow$ `[0-9]{2}-[0-9]{2}-[0-9]{4}`. Quando termos de uma propriedade p_1 de um serviço correspondem com tal $\mathcal{R}$, então p_1 pode ser considerada uma propriedade equivalente de `dbo:date`.

O *Properties Matcher* é o componente central do *Engine*, responsável por realizar o *matching* do conjunto de termos de cada propriedade fornecida por um serviço de dados, com o conjunto de termos de cada propriedade existente nas fontes externas. O *matching* é baseado na sobreposição de dados existentes entre propriedades em diferentes *datasets*. A seguinte equação foi elaborada para calcular a força (valor de 0 a 1) do *match* entre duas propriedades p_1 e p_2:

$$\mathcal{F}(p_1, p_2) = \frac{|\mathcal{V}p_1 \bigcap_s \mathcal{V}p_2|}{|\mathcal{V}p_1|} \quad (1)$$

onde $\mathcal{V}p_1$ é o conjunto de termos providos pelo serviço de dados para uma dada propriedade p_1, e $\mathcal{V}p_2$ é o conjunto de valores de uma propriedade p_2 existente em uma fonte externa. A intersecção $\bigcap_s$ utiliza um algoritmo baseado em similaridade para verificar se um termo $t_1 \in \mathcal{V}p_1$ é similar a um termo $t_2 \in \mathcal{V}p_2$ para, então, ser considerado como uma correspondência válida. Para fins de simplificação, neste trabalho foi empregada a distância de Levenshtein, embora outras medidas de similaridade possam ser incorporadas à abordagem. Com base na força da sobreposição de termos, é possível identificar o grau de correspondência entre duas propriedades. Portanto, quanto maior a força, maior a probabilidade de que as propriedades sejam equivalentes.

OntoGenesis API. É uma API RESTful destinada a ser acessível por serviços de dados que desejam ser enriquecidos semanticamente. Esta API expõe duas funcionalidades principais: i) registro de serviços de dados na arquitetura e ii) invocação do OntoGenesis *Engine* para enriquecer semanticamente as representações recebidas dos serviços de dados registrados. Assim, quando um determinado consumidor interage com um serviço registrado, este envia à API – através do *Semantic Adapter* – o recurso solicitado. O OntoGenesis API direciona a representação ao OntoGenesis *Engine* e retorna ao serviço sua nova ontologia, juntamente com os mapeamentos semânticos. Portanto, serviços de dados legados que fornecem representações sintáticas podem se registrar no OntoGenesis e ser dinamicamente enriquecidos com conceitos semânticos.

Semantic Adapter. Trata-se de um adaptador que se conecta ao serviço de dados no momento da sua implantação. Conforme mostrado na Figura 1, ele é responsável pela interação entre o serviço de dados e o OntoGenesis API. Ao usar este componente, o registro do serviço é executado automaticamente quando iniciado. Além disso, o *Semantic Adapter* intercepta todas as requisições dos consumidores que chegam ao serviço e invoca, de forma transparente, o OntoGenesis API, que delega ao *Engine* a construção de uma ontologia de domínio e mapeamentos semânticos. Com base nestes artefatos, uma nova representação semântica serializada em JSON-LD é gerada pelo adaptador e retornada ao cliente. Portanto, em vez de prover uma representações puramente sintáticas, o serviço passa a fornecer aos consumidores dados descritos semanticamente.

4.3 Resultados Preliminares

Um experimento preliminar foi realizado com o objetivo de avaliar a aplicabilidade da proposta em um cenário com dados reais. Para tanto, foi construído um serviço que provê dados governamentais atinentes a BOs publicados pela SSP/SP[1]. Como fontes externas, foram utilizadas ontologias e *subsets* do DBpedia[2] e do Geonames[3]. O serviço de dados provê informações de pessoas envolvidas em um BO, tais como nome, RG, data de nascimento, local de nascimento, etc. Já os dados obtidos do DBpedia fornecem informações de pessoas contidas no Wikipedia, enquanto os dados do Geonames se referem à locais do Brasil, bem como nome de países. Tais dados foram selecionados por ter um potencial de intersecção de dados.

Foram definidos inicialmente dois *thresholds* para a força da equivalência de propriedades: 0.5 e 0.8. Desse modo, a cada requisição feita pelo consumidor, o serviço envia a sua representação ao OntoGenesis, o qual extrai todos os valores, executa o *matching* dos termos com as fontes externas e, finalmente, cria a ontologia (se for a primeira requisição feita pelo serviço) ou a atualiza com novas propriedades equivalentes (`owl:equivalentProperty`) de acordo com o *threshold* configurado. Portanto, apenas propriedades equivalentes que atingiram o *threshold* serão incluídas na ontologia.

Os experimentos consistiram em sortear e requisitar ao serviço de dados a informação de 100 pessoas, repetindo este processo 7 vezes para cada *threshold*. Cada requisição realizada equivale a uma pessoa retornada pelo serviço, totalizando 100 requisições a cada repetição do experimento. Foi utilizada uma máquina equipada com processador Intel i7 de 2.5GHz, com 8GiB de RAM e Oracle JDK 8.

[1] http://www.ssp.sp.gov.br/transparenciassp/
[2] Datasets disponíveis em: http://wiki.dbpedia.org/Downloads2015-10
[3] Datasets disponíveis em: http://download.geonames.org/export/dump/

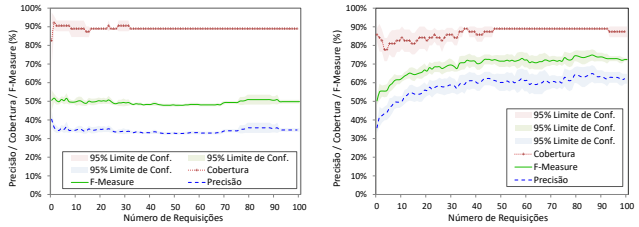

Figura 3: Precisão, Cobertura e F-Measure utilizando *threshold* da força de equivalência (a) α = 0.5 e (b) α = 0.8.

A Figura 3 apresenta a média de precisão, cobertura (*recall*) e F-Measure por requisição de ambos *thresholds*, considerando as 7 rodadas de execução e seu intervalo de confiança na área sombreada ao redor das linhas. A Figura 3 (a) mostra que no cenário com um limite de força de 0.5, o *recall* atinge quase 0.9 e mantém-se estável após 30 requisições, enquanto a precisão diminui levemente para 0.3. Por outro lado, no cenário com *threshold* de 0.8, as medidas aumentam significativamente, tendo uma precisão de quase 0.6, com *recall* e F-Measure de aproximadamente 0.85 e 0.7, respectivamente.

Adicionalmente, foi realizado um experimento com dois *matchers* extensionais, PARIS [21] e AROMA [8], com o objetivo de compará-los com a nossa abordagem. Nesse experimento, a ontologia gerada pelo OntoGenesis (sem alinhamentos) incluindo 1021 instâncias de pessoas foi alinhada com uma ontologia combinando classes, propriedades e instâncias referentes às fontes externas do experimento anterior. O tempo de alinhamento para cada *matcher* é apresentado na Figura 4. Visto que os *matchers* extensionais se alicerçam na coocorrência de indivíduos entre duas ontologias, nenhuma correspondência foi obtida, diferentemente da nossa proposta em que o *matching* se baseia no compartilhamento das propriedades.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o OntoGenesis, uma arquitetura para enriquecimento semântico de serviços de dados de forma automática. No geral, os resultados da avaliação são promissores e mostram que o OntoGenesis pode enriquecer progressivamente com semântica os serviços de dados, otimizando a reutilização de conceitos bem conhecidos de fontes externas de dados. Com isso, o serviço que antes fornecia dados sintáticos, passa a prover dados semânticos, alcançando benefícios em diversas áreas de aplicação, dando ensejo, por exemplo, à realização de novas inferências sobre dados governamentais, além de facilitar a integração com outras fontes relevantes. Espera-se, pois, que este projeto contribua para as áreas de Web *Services* e Web Semântica, e, sobretudo, diminua o tempo e o esforço demandados para a construção de serviços Web semânticos.

O trabalho encontra-se na etapa 4 da metodologia (seção 4.1). Como trabalho futuro, será considerada, na equação de força, a frequência de cada termo de modo a minimizar os falsos positivos sem introduzir os falsos negativos. Também é interessante desenvolver um mecanismo para identificar equivalência de classes, além de propriedades. Planeja-se, ainda, avaliar o uso do OntoGenesis integrado a outras ferramentas e em outros cenários, utilizando novos *datasets* governamentais e/ou científicos.

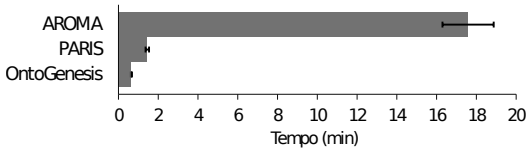

Figura 4: Comparação dos tempos de processamento.

REFERÊNCIAS

[1] Auhood Alfaries. 2010. *Ontology Learning for Semantic Web Services.* Ph.D. Dissertation. Brunel University, UK. http://dspace.brunel.ac.uk/handle/2438/4667

[2] Sean Bechhofer, Frank Van Harmelen, Jim Hendler, Ian Horrocks, Deborah L McGuinness, Peter F Patel-Schneider, and Lynn Andrea Stein. 2004. OWL Web Ontology Language Reference. *W3C Recommendation* 10 (2004). http://www.w3.org/TR/owl-ref/

[3] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American* 284, 5 (2001), 34–43.

[4] Devis Bianchini, Valeria De Antonellis, and Michele Melchiori. 2015. Developers' Networks Contribution to Web Application Design. In *Proc. of the 17th Internat. Conf. on Information Integration and Web-based Applications & Services (iiWAS '15).* ACM, New York, NY, USA, 55:1–55:10.

[5] Maricela Bravo, José Rodríguez, and Jorge Pascual. 2014. SDWS: Semantic Description of Web Services. *International Journal of Web Services Research* 11, 2 (2014), 1–23.

[6] Michael J Carey, Nicola Onose, and Michalis Petropoulos. 2012. Data Services. *Communications of the ACM* 55, 6 (jun 2012), 86.

[7] Philipp Cimiano and Johanna Völker. 2005. Text2Onto: A Framework for Ontology Learning and Data-driven Change Discovery. In *Proc. of the 10th Internat. Conf. on Natural Language Processing and Information Systems (NLDB'05).* Springer-Verlag, Berlin, Heidelberg, 227–238.

[8] Jérôme David. 2007. Association Rule Ontology Matching Approach. *Int. Journal on Semantic Web and Information Systems* 3, 2 (2007), 27–49.

[9] Jérôme Euzenat and Pavel Shvaiko. 2007. *Ontology Matching.* Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[10] Aissa Fellah, Mimoun Malki, and Atilla Elçi. 2016. Web Services Matchmaking Based on a Partial Ontology Alignment. *Int. Journal of Information Technology and Computer Science (IJITCS)* 8, 6 (June 2016), 9–20.

[11] Barbara Kitchenham. 2004. *Procedures for Performing Systematic Reviews.* Technical Report TR/SE-0401. Department of Computer Science, Keele University.

[12] Cleber Lira and Paulo Caetano. 2016. REST-Based Semantic Annotation of Web Services. In *Information Technology: New Generations*, Vol. 448. Springer, 269–279.

[13] Chengduo C.a Luo, Zibin c Zheng, Xiaorui X.d Wu, F.d Fei Yang, and Yao Y.a Zhao. 2016. Automated structural semantic annotation for RESTful services. *Internat. Journal of Web and Grid Services* 12, 1 (2016), 26–41.

[14] Alexander Maedche and Steffen Staab. 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16, 2 (March 2001), 8.

[15] S.a. McIlraith, T.C. Son, and Honglei Zeng Honglei Zeng. 2001. Semantic Web Services. *IEEE Intelligent Systems* 16, 2 (2001), 46–53.

[16] Thi Thanh Sang Nguyen and Haiyan Lu. 2016. Domain Ontology Construction Using Web Usage Data. In *Advances in Artificial Intelligence.* Springer, 338–344.

[17] Bruno C. N. Oliveira, Ivan Salvadori, Alexis Huf, and Frank Siqueira. 2016. A platform to enrich, expand and publish linked data of police reports. *Proceedings of the 15th International Conference WWW/Internet 2016*, 111–118.

[18] Silvia Quarteroni, Marco Brambilla, and Stefano Ceri. 2013. A bottom-up, knowledge-aware approach to integrating and querying web data services. *ACM Transactions on the Web* 7, 4 (2013), 19:1–19:33.

[19] Sara Salem and Samir AbdelRahman. 2010. A Multiple-domain Ontology Builder. In *Proc. of the 23rd Internat. Conf. on Computational Linguistics.* Association for Computational Linguistics, Stroudsburg, USA, 967–975.

[20] Ivan Luiz Salvadori, Alexis Huf, Bruno C. N. Oliveira, Ronaldo Santos Mello, and Frank Siqueira. 2017. Improving Entity Linking with Ontology Alignment for Semantic Microservices Composition. *Internat. Journal of Web Information Systems* 13 (2017). Issue 3.

[21] Fabian M Suchanek, Serge Abiteboul, and Pierre Senellart. 2011. Paris: Probabilistic alignment of relations, instances, and schema. *Procedures of the VLDB Endowment* 5, 3 (2011), 157–168.

[22] Davide Tosi and Sandro Morasca. 2015. Supporting the semi-automatic semantic annotation of web services: A systematic literature review. *Information and Software Technology* 61 (may 2015), 16–32.

[23] Yuangang Yao, Hui Liu, Jin Yi, Haiqiang Chen, Xianghui Zhao, and Xiaoyu Ma. 2014. An automatic semantic extraction method for web data interchange. *2014 6th Int. Conf. on Computer Science and Information Technology* (2014), 148–152.

[24] Zhen Zhang, Shizhan Chen, and Zhiyong Feng. 2013. Semantic Annotation for Web Services Based on DBpedia. In *Proc. of the IEEE Seventh International Symposium on Service-Oriented System Engineering (SOSE '13).* 280–285.