Semantic Enrichment of Services for Linked Data Provision in SOA

Enriquecimento Semântico de Serviços para Provimento de Dados Conectados em SOA

Bruno C. N. Oliveira
bruno.cno@posgrad.ufsc.br
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Santa Catarina, Brazil

Alexis Huf
alexis.huf@posgrad.ufsc.br
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Santa Catarina, Brazil

Ivan Salvadori
ivan.salvadori@posgrad.ufsc.br
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Santa Catarina, Brazil

Frank Siqueira
frank.siqueira@ufsc.br
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Santa Catarina, Brazil

ABSTRACT

One of the main challenges concerning information systems that deal with heterogeneous services and data consists in achieving semantic interoperability. Several approaches have been proposed to assist in the semantic enrichment of services, by using domain ontologies. Nevertheless, most of the proposals found in the literature focus on the enrichment of the service description, neglecting data representations provided by such services. This work presents an approach to dynamically provide representations in the form of linked data in a service-oriented system. Therefore, we propose an architecture to build/evolve ontologies and to enrich services with concepts defined by such ontologies. As a way to increase the semantic expressiveness of services, the architecture employs ontology matching techniques to explore equivalences in external sources. The results show that the proposed solution can achieve satisfactory precision and coverage levels, as well as better performance compared to other approaches.

CCS CONCEPTS

• **Information systems → Web services**; *Ontologies*; • **Applied computing → Information integration and interoperability**;

KEYWORDS

Service-Oriented Architecture, Interoperability, Semantic Web, Linked Data, Ontology Construction and Matching.

SBSI'19, May 20–24, 2019, Aracaju, Brazil

ACM ISBN 978-1-4503-7237-4/19/05…$15.00
https://doi.org/10.1145/3330204.3330258

ACM Reference Format:
Bruno C. N. Oliveira, Alexis Huf, Ivan Salvadori, and Frank Siqueira. 2019. Semantic Enrichment of Services for Linked Data Provision in SOA. In *XV Brazilian Symposium on Information Systems (SBSI'19), May 20–24, 2019, Aracaju, Brazil.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3330204.3330258

1 INTRODUÇÃO

Serviços de dados – ou seja, serviços que manipulam e fornecem informações oriundas de alguma fonte de dados – têm sido cada vez mais utilizados em decorrência da popularização das arquiteturas orientadas a serviços (SOA). Em um ambiente com sistemas de informação heterogêneos que empregam SOA, a integração dos serviços é um requisito fundamental para alcançar a interoperabilidade. Nesse sentido, Boscarioli et al. [3] evidenciam a importância de adotar formatos de dados sofisticados, em especial os oriundos da Web semântica, para assegurar a interoperabilidade da informação. As soluções de interoperabilidade semântica destacam-se por utilizar ontologias, permitindo a troca de informações por meio de conceitos compartilhados [3].

Serviços de dados podem empregar as tecnologias da Web semântica [1] (sendo assim denominados serviços Web semânticos) com o intuito de prover dados conectados (*Linked Data*) [2] e, por conseguinte, facilitar a reutilização e a integração com sistemas mais complexos. Promovendo o reúso em SOA, uma organização tende a diminuir custos de desenvolvimento e a aumentar a qualidade das suas aplicações [16]. Ademais, diversos pesquisadores apontam os benefícios de se empregar serviços semânticos, não apenas na interoperabilidade dos dados [18, 20], mas também para automatizar processos como descoberta, seleção e composição de serviços [13].

Embora a literatura apresente diversos benefícios oriundos do uso de serviços semânticos, a sua efetiva implementação e adoção ainda são limitadas. Em geral, as fontes de dados acessadas pelos serviços armazenam os dados sem explicitar a sua semântica, o que dificulta o reúso e a integração eficaz desses serviços. Com isso, faz-se necessário enriquecer semanticamente os serviços construindo

uma ontologia de domínio que descreva os dados, além de estabelecer as associações semânticas entre a ontologia e o serviço que manipula tais dados. Não obstante, em ambientes modernos que empregam SOA, diferentes ontologias podem coexistir, tornando necessário o uso de técnicas que permitam alinhar os conceitos das ontologias visando o reúso e a interoperabilidade entre os serviços. Isso traz um novo desafio para o desenvolvimento e a integração de serviços, sobretudo pelo alto grau de complexidade e pelo tempo despendido para construir ontologias e realizar o *match* dos diferentes conceitos existentes. Além disso, a eficácia das ferramentas disponíveis para construção e *matching* de ontologias depende da disponibilidade de um volume significativo de dados, o que nem sempre é possível quando o acesso aos serviços é feito sob demanda.

Analisando os trabalhos propostos na literatura acerca do enriquecimento semântico de serviços em um ambiente SOA, observou-se que, além de requererem um alto grau de intervenção humana, as abordagens propostas focam, majoritariamente, na geração de descrições semânticas das interfaces dos serviços [4, 10]. Os trabalhos que buscam enriquecer as representações dos dados fornecidas pelos serviços de modo a prover dados conectados, requerem que as ontologias já tenham sido construídas [18], ou, ainda, não consideram o enriquecimento do serviço em si de modo dinâmico [7]. Ademais, os trabalhos que abordam a construção de ontologias [22] não preveem o seu alinhamento com outras ontologias externas ou exigem uma grande massa de dados como entrada [15].

Para superar tais desafios, este trabalho apresenta uma abordagem para enriquecer dinamicamente serviços de dados, de tal modo que estes sejam capazes de fornecer descrições e representações semânticas na forma de dados conectados. Baseado nessa abordagem, propomos uma arquitetura, denominada OntoGenesis, a qual pode ser implantada em um ambiente SOA já existente com o objetivo de construir e evoluir ontologias de domínio que descrevem os dados fornecidos pelos serviços, identificando equivalências entre as propriedades da ontologia criada com propriedades existentes em ontologias externas. Tais equivalências são adicionadas à ontologia, e tanto a descrição quanto as representações fornecidas pelo serviço são associadas aos conceitos dessa ontologia.

Como contribuição, a arquitetura proposta busca abstrair, do ponto de vista dos desenvolvedores de *software*, o processo oneroso de enriquecer semanticamente serviços que fornecem dados sem informação semântica, provendo dinamicamente dados conectados e fomentando a interoperabilidade entre sistemas. A proposta foi avaliada utilizando dados abertos de publicações científicas, disponibilizados pelo DBLP, bem como *datasets* e ontologias públicas do ScholarlyData, DublinCore e FOAF. Os experimentos conduzidos mostram a aplicabilidade da proposta, e os resultados obtidos apresentam níveis adequados de conformidade e de desempenho.

O restante deste artigo está organizado em 6 seções. A seção 2 apresenta os conceitos fundamentais para compreensão do trabalho. A seção 3 apresenta uma visão geral da abordagem de enriquecimento semântico de serviços, enquanto a arquitetura proposta é detalhada na seção 4. A seção 5 expõe os experimentos e analisa os resultados obtidos. A seção 6 discute os principais trabalhos relacionados e, por fim, a seção 7 apresenta as conclusões e direcionamentos para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SOA e a Web Semântica

Sistemas de informação que possuem uma arquitetura orientada a serviços têm sido comumente adotados pela indústria e pela comunidade acadêmica, em decorrência dos seus benefícios perante às arquiteturas monolíticas. SOA pode ser entendida como um paradigma para organizar e (re)utilizar recursos distribuídos que podem estar sob diferentes domínios [11]. Joyce et al. [16] observam que há fatores importantes que influenciam na eficácia do reúso de serviços em ambientes SOA, como o custo, a existência de sistemas legados e ambientes heterogêneos. Em cenários que envolvem serviços independentes e heterogêneos, a interoperabilidade é considerada como um dos maiores desafios a serem alcançados, perpassando contextos de IoT, ecossistemas de software e *cloud computing* [3].

De acordo com a OASIS [11, p. 8], a descrição dos serviços (que dispõe de informações sobre o que ele oferece e as suas capacidades) deve ser feita em um formato cuja sintaxe e semântica sejam amplamente acessíveis e compreensíveis, facilitando a interoperabilidade entre sistemas e dados. As tecnologias da Web Semântica [1], como dados conectados e ontologias, surgem como um meio de prover significado aos dados, promovendo a sua compreensão não apenas por humanos, mas também por máquinas. Em [3, cap. 5] é possível observar uma tendência em direção a manter dados e serviços amplamente disponíveis e conectados. No esteio da Web Semântica, os dados conectados desempenham um papel fundamental para a publicação, compartilhamento e ligação de dados [2]. Essa interconexão de informações advém da combinação de conteúdos – que, em um ambiente SOA, são disponibilizados por serviços – com descrições semânticas de seus significados e relacionamentos.

Os serviços semânticos, portanto, empregam as tecnologias da Web Semântica com o intuito de prover informações semanticamente enriquecidas, promovendo a sua compreensão por outros agentes de *software*, o que, em última análise, tende a melhorar a capacidade de reutilização e a interoperabilidade dos serviços. Para tanto, um serviço enriquecido semanticamente pode fornecer dados conectados a recursos descritos por ontologias.

2.2 Construção e *Matching* de Ontologias

As ontologias desempenham um papel fundamental na Web Semântica, especialmente no âmbito de Web *Services*. Em geral, engenheiros de ontologia as desenvolvem manualmente buscando fornecer um modelo específico de domínio adequado para descrever a semântica dos dados geridos pelos serviços pertencentes a um sistema de informação. O maior esforço envolvido durante o processo de enriquecimento semântico dos serviços está na construção das ontologias, bem como na sua manutenção e evolução. *Ontology Learning* (OL) [12] é uma área de pesquisa que busca automatizar a construção de ontologias, extraindo os conceitos e relações de diferentes recursos de forma (semi-)automática.

Embora a OL ofereça mecanismos para automatizar o processo de construção de ontologias, é essencial que conceitos existentes em diferentes ontologias sejam reutilizados, a fim de ampliar as possibilidades de integração. Neste sentido, as técnicas de *matching* de ontologia surgem para resolver questões de heterogeneidade de ontologias, identificando as correspondências – geralmente expressas por relações de equivalência – entre ontologias distintas. Euzenat

e Shvaiko [6] identificam quatro técnicas para *matching* de ontologias: i) baseada em nomes, que considera apenas os rótulos dos elementos da ontologia; ii) estrutural, que observa a estrutura dos elementos ontológicos; iii) baseada em semântica, a qual executa *reasoning* para inferir equivalências; e iv) extensional, que utiliza as instâncias das ontologias para encontrar indivíduos equivalentes e assim estabelecer as correspondências.

Visto que os serviços de um sistema de informação complexo manipulam e fornecem diversos tipos de informação, 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 equivalências entre as propriedades das ontologias criadas para os serviços e ontologias de fontes externas já existentes. Tais equivalências são denotadas pelo axioma `owl:equivalentProperty` da *Web Ontology Language* (OWL) [14].

3 ENRIQUECIMENTO SEMÂNTICO DE SERVIÇOS DE DADOS

Este trabalho propõe uma abordagem para fornecer dinamicamente dados conectados em um sistema que emprega uma arquitetura orientada a serviços. Mais especificamente, a abordagem enriquece semanticamente serviços de dados de modo a gerar em tempo de execução tanto uma descrição do serviço quanto representações semânticas dos dados. As representações dos dados são fornecidas aos consumidores dos serviços (usuários e/ou outros agentes de *software*) em formato de dados conectados. Para isso, é necessário um mecanismo capaz de construir e evoluir ontologias de domínio a partir de representações sintáticas (ou seja, sem informação semântica) disponibilizadas pelo serviço aos seus consumidores. Alinhado a isso, um adaptador semântico (referenciado ao longo do artigo como *Semantic Adapter*) deve ser incorporado ao serviço a fim de viabilizar as associações semânticas e, por conseguinte, fornecer aos consumidores novas representações, bem como uma nova descrição da interface do serviço, sem nenhum impacto nos demais serviços que compõem o sistema.

A Figura 1 apresenta uma visão geral da abordagem para enriquecimento semântico de um serviço de dados. Inicialmente, um consumidor envia uma requisição para o serviço. A requisição é interceptada pelo *Semantic Adapter*, que envia a um *Enricher* uma representação sintática (serializada, por exemplo, em XML ou JSON) do dado requerido. O *Enricher* extrai todos os elementos da representação sintática e constrói uma ontologia de domínio para o serviço, incluindo classes, propriedades de dados (*datatype properties*) e propriedades de objetos (*object properties*), bem como links de propriedades equivalentes identificadas em ontologias já existentes. Tais equivalências podem ser descobertas utilizando fontes externas que possuem relações com o domínio da ontologia gerada.

Como saída, o *Enricher* retorna a ontologia de domínio criada, juntamente com associações semânticas – denominadas Mapeamentos Semânticos ($\mathcal{MS}$) – entre os atributos sintáticos das representações do serviço e os conceitos da nova ontologia. Um mapeamento semântico é definido como uma tripla $\mathcal{MS} = \{a, c, t\}$, onde a é o atributo oriundo da representação sintática, c é o conceito semântico representado na ontologia e t é seu tipo: uma classe, uma propriedade de dados ou uma propriedade de objetos. Suponha que tenha sido criada uma propriedade de dados c, onde c = `"http://servico1/ontology#name"`, para o atributo a = `"name"` de uma determinada representação. O mapeamento semântico gerado deve ser $\mathcal{MS}$ = {`"name"`, `"http://servico1/ontology#name"`, `"Datatype-property"`}.

Os mapeamentos semânticos são úteis para gerar não apenas representações, mas também descrições semânticas do serviço. A descrição semântica é gerada de acordo com os parâmetros de entrada e saída do serviço e é disponibilizada pelo *Semantic Adapter* de modo que os clientes semânticos possam consumí-la e, então, interpretar as capacidades do serviço. É importante ressaltar que, caso um consumidor não tenha suporte a formatos de dados conectados, como JSON-LD (JSON *for Linked Data*) [9], o serviço mantém o envio de dados sintáticos, não afetando consumidores legados.

Conforme mostrado na Figura 1, o *Semantic Adapter* é um componente conectado a um serviço de dados que tem como objetivo registrar os serviços no *Enricher* e interceptar o retorno das requisições para fornecer dados conectados aos consumidores. O JSON-LD é utilizado para serializar não apenas as representações fornecidas pelo serviço, mas também a sua descrição, a qual define as entidades que o serviço gerencia e provê informações de como obtê-las. O *Semantic Adapter* pode ser visto como um conector responsável pela comunicação entre o serviço e o *Enricher*. A seção 4 apresenta a arquitetura do *Enricher* e o funcionamento do *Semantic Adapter*.

4 ARQUITETURA PROPOSTA

Em conformidade com a abordagem dinâmica apresentada na seção anterior, este trabalho propõe uma arquitetura, intitulada OntoGenesis, composta de componentes e subcomponentes que interagem entre si visando ao enriquecimento semântico de serviços de dados de forma automática. A arquitetura é dividida em três componentes principais, conforme ilustrado na Figura 2. O primeiro, o OntoGenesis *Engine*, é responsável pela construção/evolução de uma ontologia para o serviço de dados e pela criação dos mapeamentos semânticos de acordo com os elementos das representações sintáticas do serviço. Ele utiliza ainda fontes de dados externas já enriquecidas semanticamente para identificar equivalências e aprimorar a ontologia de domínio construída. O segundo componente, chamado OntoGenesis API, é uma Web API que fornece uma interface de comunicação para acessar as funcionalidades fornecidas pelo *Engine*. Finalmente, o *Semantic Adapter* trata-se de

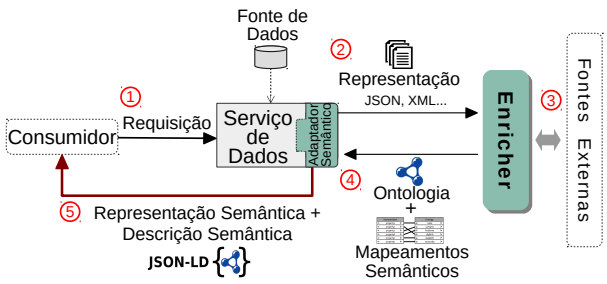


Figura 1: Método de enriquecimento dinâmico do serviço.

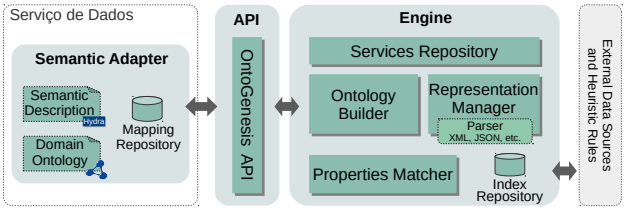


Figura 2: Visão geral da arquitetura do proposta.

uma biblioteca para acessar a OntoGenesis API e interceptar requisições, provendo dinamicamente uma descrição e representações semânticas do serviço.

É importante notar que o *Engine*, conjuntamente com a API, correspondem ao *Enricher* da abordagem ilustrada na Figura 1. As próximas seções apresentam as funções de cada componente da arquitetura e as suas interações.

4.1 OntoGenesis Engine

Conforme ilustrado na Figura 2, o OntoGenesis *Engine* é composto por outros sub-componentes, descritos seguir.

O *Services Repository* gerencia informações sobre os serviços de dados registrados, como o nome do serviço, sua localização 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 incorporados, 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 3 ilustra uma amostra de uma ontologia (em Turtle) (b) com base em uma dada representação em XML (a). Os valores contidos no XML representam dados reais abertos de publicações científicas disponibilizados pelo DBLP[1].

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

O *Index Repository* armazena, em uma base de dados NoSQL do tipo chave-valor, índices de dados literais originários tanto de fontes externas quanto dos serviços para os quais o OntoGenesis

[1] *Dataset* do DBLP: https://dblp.uni-trier.de/xml/.

```xml
<dblp>
  <article key="journals/ijswis/BizerHB09">
    <author orcid="0000-0003-2367-0237">Christian Bizer</author>
    <author>Tom Heath</author>
    <author>Tim Berners-Lee</author>
    <title>Linked Data - The Story So Far</title>
    <pages>1-22</pages>
    <year>2009</year>
    <volume>5</volume>
    <journal>Int. J. on Semantic Web and Inf. Syst.</journal>
    ...
  </article>
  ...
</dblp>
```

(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 .
:Dblp       a owl:Class .
:Article    a owl:Class .
:temArticle a owl:ObjectProperty;     rdfs:domain  :Dblp;
                                      rdfs:range   :Article .
:author a owl:DatatypeProperty ;      rdfs:domain  :Article;
                                      rdfs:range   xsd:string .
:key a owl:DatatypeProperty ;         rdfs:domain  :Article;
                                      rdfs:range   xsd:string .
:journal a owl:DatatypeProperty ;     rdfs:domain  :Article;
                                      rdfs:range   xsd:string .
:year a owl:DatatypeProperty ;        rdfs:domain  :Article;
                                      rdfs:range   xsd:date .  # ...
```

(b)

Figura 3: Amostra de: (a) Representação em XML de um artigo; e (b) Ontologia construída a partir da representação.

construirá as ontologias de domínio. Dois tipos de índice foram projetados: um para indexar as informações dos dados fornecidos pelo serviço e outro para indexar os dados extraídos de fontes externas. Considerando que os conjuntos de dados possuem propriedades e valores, ambos os índices usam a propriedade como chave para indexação do valor (ou objeto, no caso das fontes externas). 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 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 busca calcular a força (valor de 0 a 1) da equivalência 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.

4.2 OntoGenesis API

É uma API RESTful 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 construção da ontologia e enriquecimento semântico das representações. Quando um determinado consumidor interage com um serviço registrado, o *Semantic Adapter* envia à API o recurso solicitado pelo cliente. A OntoGenesis API direciona a representação ao *Engine* e retorna ao serviço sua nova ontologia, juntamente com os mapeamentos semânticos. Assim, serviços legados que fornecem representações sintáticas podem se registrar na OntoGenesis e ser dinamicamente enriquecidos com conceitos semânticos.

A OntoGenesis API suporta configurações personalizadas, como o *threshold* para a força de equivalência de propriedades (α, com valor de 0 a 1) e um tamanho de *buffer* de representações. O tamanho do *buffer* indica quantas representações fornecidas por um serviço devem ser enviadas à OntoGenesis *Engine* concomitantemente. Ademais, a OntoGenesis API oferece uma interface que permite carregar novas fontes de dados externas no *Index Repository*. Consequentemente, os usuários podem carregar novos *datasets*, para que seus dados sejam considerados em tempo de execução pelo *Properties Matcher* em futuras requisições dos serviços de dados.

4.3 Semantic Adapter

Trata-se de um adaptador que se conecta ao serviço no momento da sua implantação. Conforme mostrado na Figura 2, ele é responsável pela interação entre os serviços e a 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, a OntoGenesis API, que delega ao *Engine* a construção de uma ontologia de domínio e de mapeamentos semânticos (armazenados no próprio componente). Com base nestes artefatos, gera-se uma descrição baseada no vocabulário Hydra [8], bem como uma nova representação semântica serializada em JSON-LD, a qual é retornada ao cliente. O Hydra busca simplificar o desenvolvimento de descrições serviços fazendo uso dos benefícios providos pelos dados conectados. Uma vantagem do Hydra é a possibilidade de especificar uma série de conceitos comumente utilizados nos serviços, dando ensejo à criação de clientes genéricos.

Para criar as representações na forma de dados conectados, o *Semantic Adapter* faz uso dos tokens `@id`, `@type` e `@context` do JSON-LD [9]. Os atributos do JSON-LD são mantidos da mesma forma que declarados na representação sintática, de modo a ter pouco impacto nos nomes dos campos presentes na representação original. Um dos principais benefícios dessa estratégia é permitir

```
{ "@context": {
      "Article": "http://exemplo-servico/ontology/Article",
      "key": "http://exemplo-servico/ontology/key",
      "author": "http://exemplo-servico/ontology/author",
      "title": "http://exemplo-servico/ontology/title",
      "pages": "http://exemplo-servico/ontology/pages",
      "year": "http://exemplo-servico/ontology/year",
      ...
      },
      "@id": "http://exemplo-servico/article?key=journals/ijswis/BizerHB09",
      "@type": "Article", ...
}
```

Figura 4: Excerto de um contexto JSON-LD

que não apenas os consumidores semânticos, mas também os não semânticos (ou seja, aqueles que conseguem processar apenas representações sem nenhuma informação semântica) processem as novas saídas do serviço, uma vez que os rótulos dos atributos se mantêm inalterados. O trecho de código da Figura 4 mostra um contexto JSON-LD mapeando os atributos da representação XML com os conceitos da ontologia ilustrados no exemplo da Figura 3.

5 AVALIAÇÃO E RESULTADOS

Com o objetivo de avaliar a aplicabilidade da proposta e analisar medidas de conformidade (como precisão – que visa medir quantas equivalências estão corretas, dentre todas obtidas – e cobertura, que representa a fração das equivalências relevantes obtidas, dentre todas as equivalências existentes) e desempenho (tempo de processamento e consumo de memória), foram conduzidos experimentos em um cenário com dados reais coletados a partir de diferentes fontes abertas.

Para tanto, o OntoGenesis foi implantada em um ambiente contendo um serviço de dados que fornece representações sintáticas referentes a metadados de publicações científicas, tais como autores, título, ano, nome do periódico/evento, dentre outros. Os dados fornecidos pelo serviço correspondem a um subconjunto dos dados disponibilizados pelo DBLP em formato XML, sem qualquer informação semântica. Como fontes externas, foram utilizados os vocabulários do FOAF e Dublin Core, bem como *datasets* abertos do ScholarlyData[2] e a ontologia que descreve os seus dados[3].

5.1 Metodologia

Foram definidos três *thresholds* α para a força da equivalência das propriedades: 0.4, 0.6 e 0.8. Desse modo, a cada requisição feita pelo consumidor, o serviço envia a representação ao OntoGenesis, o qual extrai todos os valores, executa o *matching* dos termos com as fontes externas e, finalmente, cria (se for a primeira requisição) ou atualiza a ontologia com novas propriedades equivalentes de acordo com o *threshold* configurado. Logo, apenas propriedades equivalentes que atingiram o *threshold* são incluídas na ontologia.

O serviço construído fornece ao OntoGenesis, por meio do *Semantic Adapter*, representações sintáticas que podem conter até nove atributos relacionados a uma publicação. Cinco desses possuem uma ou mais equivalências em fontes externas, enquanto os demais não possuem nenhum tipo de correspondência.

[2] *Datasets* do ScholarlyData: http://www.scholarlydata.org/dumps/.
[3] Conference Ontology: http://www.scholarlydata.org/ontology/doc/.

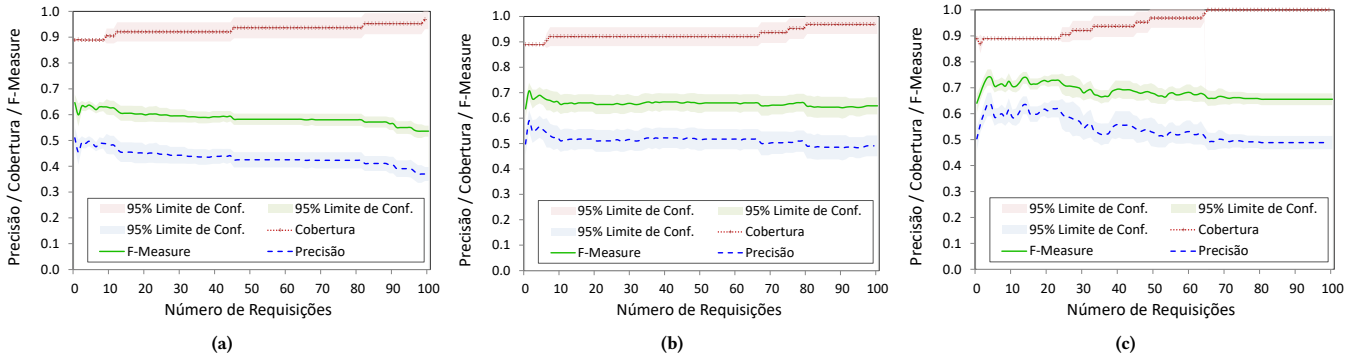


Figura 5: Precisão, Cobertura e F-Measure utilizando *threshold* da força de equivalência (a) α = 0.4; (b) α = 0.6 e (c) α = 0.8.

Os experimentos consistiram em requisitar aleatoriamente ao serviço de dados informações de 100 publicações, repetindo este processo 7 vezes para cada *threshold*. O número de rodadas demonstrou-se satisfatório, visto que os resultados a 95% de intervalo de confiança apresentaram baixa variação. Cada requisição realizada equivale a uma publicação retornada pelo serviço, totalizando 100 requisições a cada repetição do experimento. Foi utilizada uma máquina equipada com processador Intel Core i7 de 2.5GHz, com 8GiB de RAM, com sistema operacional Ubuntu 16.04 e Oracle JDK 8. Os códigos-fonte, bem como os resultados obtidos estão disponíveis em repositórios públicos[4].

5.2 Análise de conformidade e desempenho

A Figura 5 apresenta a curva média de precisão, cobertura e *F-Measure* por requisição, considerando as 7 rodadas de execução e seus intervalos de confiança de 95% na área sombreada ao redor das linhas. Com o *threshold* definido para 0.4 (Figura 5 (a)), é possível observar um decréscimo da precisão e um aumento tênue na cobertura. No cenário com $\alpha = 0.6$ (Figura 5 (b)), há uma estabilidade da precisão, alcançando níveis mais altos apenas a partir da 80ª requisição, aproximadamente. Já na Figura 5 (c) (com $\alpha = 0.8$), observa-se uma maior variação na precisão e um crescimento mais notório da cobertura, alcançando 100% com intervalo de confiança de 0 a partir da 66ª requisição.

Alguns fatores explicam a variação inicial da precisão acompanhada pela sua leve queda ilustrada na Figure 5 (c). Propriedades como `givenName` e `familyName`, tanto do FOAF quanto da *Conference Ontology*, são identificadas pelo OntoGenesis, em alguns casos, como equivalentes à propriedade `author` da ontologia criada para os dados do DBLP. Contudo, como `author` contém nomes completos, essas associações são consideradas falsos positivos, reduzindo, assim, a precisão. Apenas `foaf:name` e `cofunc:name` são aceitas como equivalentes de `author`.

Além disso, existe o fato de propriedades da ontologia gerada para os dados do DBLP possuírem mais de uma equivalência com propriedades das ontologias que descrevem o ScholarlyData. Isto se deve ao fato de os metadados das conferências no ScholarlyData estarem associados a mais de uma ontologia. Por exemplo, um autor de um artigo do ESWC é descrito pela *DatatypeProperty* `cofunc:name` da *Conference Ontology*, enquanto um autor no ICWS é descrito pela propriedade `foaf:name` do FOAF. Já outras conferências usam a propriedade `dc:creator` do Dublin Core. Ademais, existe o fato de que a propriedade `cofunc:name` pode descrever tanto o nome de um autor quanto o nome de um evento. Isto porque tal propriedade possui domínio em `Agent` – que é uma superclasse de `Organisation` e `Person` – o que tende a reduzir a precisão.

O tempo médio de processamento para cada requisição é apresentado na Tabela 1. O processamento foi dividido em quatro etapas: geração dos mapeamentos semânticos e construção da ontologia; processo de *matching* de propriedades; geração das representações em JSON-LD; e geração da descrição Hydra. Foram considerados os resultados observados nas 7 rodadas de execução, para cada valor de *threshold* α avaliado. Percebe-se que não há alteração significativa no tempo de execução entre os *thresholds* empregados. Ademais, o tempo de processamento é independente do número de requisições, pois as requisições anteriores não são reprocessadas, sendo recalculado somente o valor da força das propriedades equivalentes.

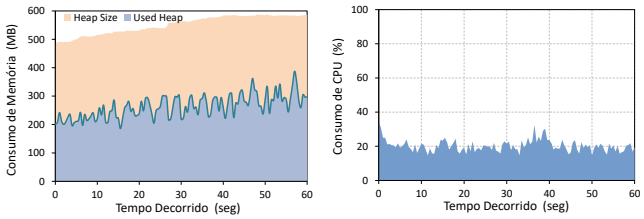
Para analisar a média de consumo de memória e CPU, a metodologia experimental foi adaptada de modo que um consumidor envie requisições aleatórias ao serviço por um período determinado. Neste caso, o tempo de execução é fixo, e não mais a quantidade de requisições. Assim, as variáveis de consumo de memória e CPU foram monitoradas durante 60 segundos após uma fase de pré-aquecimento, que objetiva descartar resultados espúrios oriundos de fatores como *caches*, inicializações e funcionalidades de otimização da JVM. Ademais, as mesmas configurações de ambiente descritas anteriormente foram usadas.

Tabela 1: Tempo de processamento (ms) com 95% de intervalo de confiança.

α*	Map. Sem. + Ontologia	*Matching* de Propriedades	Geração JSON-LD	Enriq. Descrição
0.4	19.74 ± 0.85	28.91 ± 1.77	4.15 ± 0.38	1.03 ± 0.02
0.6	20.44 ± 0.97	28.30 ± 1.98	4.04 ± 0.35	0.95 ± 0.02
0.8	20.76 ± 1.03	29.87 ± 2.19	4.25 ± 0.33	1.19 ± 0.03

*α = *Threshold* da força de equivalência

[4]https://github.com/brunocnoliveira/dblp-scholarly-ontogenesis-experiments.



(a) (b)

Figura 6: Consumo médio de (a) memória e (b) CPU.

A Figura 6 (a) e (b) apresenta o consumo médio de memória e de CPU, respectivamente, considerando as 7 rodadas de execução. Observa-se que a média do tamanho de memória *Heap* utilizada chega a quase 400MB, apresentando uma leve tendência de crescimento no consumo. Este consumo está diretamente relacionado com o tamanho do índice construído com base nos dados do ScholarlyData e do DBLP. O consumo médio de CPU manteve-se abaixo de 40% durante a execução, ficando abaixo de 20% durante parte do tempo. Visto que esse é um valor médio, foram observados, naturalmente, breves picos de uso de CPU.

5.3 Comparativo com outras abordagens

Um experimento adicional foi executado com o objetivo de comparar a abordagem proposta neste artigo com dois *matchers* de ontologia extensionais: AROMA [5] e PARIS [19], com 7 rodadas de execução para cada e utilizando o mesmo ambiente citado anteriormente. Para tanto, o cenário foi convertido de enriquecimento semântico de serviços para *matching* de ontologias. O intuito é alinhar uma ontologia construída pelo OntoGenesis contendo instâncias de pessoas (sem as triplas de equivalências `owl:equivalentProperty`) com ontologias e instâncias do DBpedia, FOAF e Geonames. Foram gerados dois arquivos: um contendo a ontologia construída pelo OntoGenesis, e outro contendo as ontologias das fontes externas juntamente com suas instâncias de dados.

Tais experimentos demonstraram que nenhum dos dois *matchers* foi capaz de identificar correspondências. A razão para a falta de correspondência consiste na ausência de compartilhamento de indivíduos (instâncias) entre os dados fornecidos pelo serviço e os dados existentes nas fontes externas. Visto que os *matchers* extensionais se alicerçam na coocorrência de indivíduos, nenhuma equivalência foi obtida, diferentemente da abordagem proposta neste artigo em que o *matching* está no compartilhamento dos valores das propriedades. Além disso, o OntoGenesis apresentou melhor desempenho: em média, o AROMA levou quase 18 minutos, o PARIS 1 minuto e 45 segundos, e o OntoGenesis cerca de 39 segundos.

6 TRABALHOS RELACIONADOS

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

As técnicas de *matching* de ontologia [6] apresentam desafios específicos no cenário de enriquecimento semântico de serviços. A técnica mais adequada para o problema que atacamos neste trabalho é o *matching* extensional, a qual pode ser usada para o enriquecimento de serviço, conforme visto em [18]. Tais técnicas, no entanto, buscam identificar correspondências entre duas ontologias a partir do compartilhamento dos seus indivíduos [5, 19]. Caso os indivíduos não compartilhem um conjunto de propriedades, o alinhamento não é bem sucedido. Antagonicamente, a proposta deste artigo, além de construir ontologias, permite gerar correspondências mesmo havendo apenas uma propriedade equivalente.

Com relação à construção de ontologias, diversas ferramentas [15, 17] foram projetadas com o objetivo de apoiar os usuários e especialistas de domínio nesse processo. Não obstante, todas elas sofrem de alguma deficiência. Primeiramente, 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 [15], diferentemente da abordagem proposta neste artigo, na qual os dados são fornecidos sob demanda pelos serviços de dados.

Yao et al. [22] apresentam um *framework* cujo objetivo é gerar uma ontologia unificada a partir de um conjunto de representações JSON. Para tanto, os elementos do JSON são convertidos em triplas RDF e um mapeamento semântico é criado de modo a obter 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 para interoperabilidade entre serviços, os autores não abordam o enriquecimento semântico automático de serviços. Ademais, a despeito de os autores não terem disponibilizado os dados utilizados nos experimentos, é possível inferir pelos resultados que a abordagem é capaz de processar 2.038 triplas/segundo no melhor caso. Em contraste, os experimentos executados com o OntoGenesis demonstraram que este foi capaz de processar aproximadamente 96.300 triplas/segundo.

Uma arquitetura para integrar serviços que utilizam diferentes tipos de fontes de dados em uma arquitetura orientada a microsserviços é proposta por Villaça e Azevedo [21]. A proposta foca em ambientes onde os dados são frequentemente atualizados. Entretanto, a arquitetura não prevê o enriquecimento semântico dos serviços, tampouco o fornecimento de dados conectados, visto que a integração é feita em um nível externo aos serviços.

Muitos pesquisadores têm envidado esforços no desenvolvimento de abordagens semi-automáticas para enriquecer serviços semanticamente [20]. Estes trabalhos podem ser divididos em dois subgrupos: i) os que tratam do enriquecimento da descrição da interface dos serviços; e ii) os que focam no enriquecimento das representações fornecidas pelo serviço, conectando-as a outros recursos semânticos. Pouca atenção, no entanto, é dada ao último grupo. A maioria das propostas encontradas na literatura tem como objetivo enriquecer as descrições do serviço [4, 10].

Salvadori et al. [18] exploram a intersecção de dados observada nas descrições dos microsserviços de um ambiente SOA com o intuito de enriquecer as representações fornecidas pelos serviços por meio de links `owl:sameAs` e `rdfs:seeAlso`. Ademais, os autores

propõem um *framework* que adota técnicas de *matching* de ontologias para identificar equivalências entre ontologias que descrevem diferentes serviços. O *framework*, contudo, 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*.

7 CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho apresentou uma abordagem para enriquecer dinamicamente sistemas em um ambiente de SOA, de tal modo que os seus serviços sejam capazes de fornecer descrições e representações associadas a conceitos semânticos. Alinhada a essa abordagem, foi proposta uma arquitetura, denominada OntoGenesis, a qual visa i) a construção e evolução de ontologias de domínio que descrevem os dados fornecidos pelos serviços, e ii) a identificação de relações de equivalências entre as propriedades da ontologia criada com propriedades existentes em ontologias externas. Diferentemente das abordagens encontradas na literatura, o OntoGenesis destina-se a enriquecer semanticamente serviços com a mínima intervenção humana. Assim, serviços que antes forneciam dados puramente sintáticos, passam a prover também dados conectados.

No geral, os resultados da avaliação são promissores e mostram que o OntoGenesis pode enriquecer semanticamente sistemas que empregam SOA, fomentando a interoperabilidade por meio da reutilização de conceitos semânticos já existentes. Como contribuição, a abordagem proposta busca facilitar o enriquecimento de sistemas orientados a serviços introduzindo elementos da Web semântica como solução de integração, diminuindo, portanto, o tempo e o esforço demandados para construção de serviços semânticos.

A principal ameaça à validade deste trabalho é a avaliação restrita ao cenário de publicações acadêmicas. Embora uma avaliação anterior tenha obtido bons resultados em um domínio de localidades geográficas e informações pessoais, não é possível afirmar que a proposta independe do domínio de aplicação ou que os resultados serão sempre satisfatórios.

Como trabalhos futuros, planeja-se refinar a equação da força de equivalência considerando a frequência de cada termo, a fim de minimizar os falsos positivos e maximizar os resultados. Além disso, visto que a proposta atual considera apenas equivalências entre propriedades, trabalhos posteriores podem explorar equivalências entre as classes das ontologias. Pretende-se, ainda, desenvolver um mecanismo para filtrar fontes de dados específicas, durante a busca de equivalências, em consonância com o domínio do serviço requisitado, com o intuito de obter melhores resultados realizando o *match* apenas com as fontes externas pertencentes ao mesmo domínio do serviço.

AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web. *Scientific American* 284, 5 (2001), 34–43.
[2] C. Bizer, T. Heath, and T. Berners-Lee. 2009. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.* 5, 3 (2009), 1–22.
[3] Clodis Boscarioli, Renata M Araujo, and Rita S P Maciel. 2017. *I GranDSI-BR Grand Research Challenges in Information Systems in Brazil 2016-2026.* Special Committee on Information Systems (CE-SI). Brazilian Computer Society (SBC).
[4] Marco Cremaschi and Flavio De Paoli. 2017. Toward Automatic Semantic API Descriptions to Support Services Composition. In *Service-Oriented and Cloud Computing*, Flavio De Paoli, Stefan Schulte, and Einar Broch Johnsen (Eds.). Springer, Cham, Switzerland, 159–167.
[5] Jérôme David. 2007. Association Rule Ontology Matching Approach. *International Journal on Semantic Web and Information Systems* 3, 2 (2007), 27–49.
[6] Jérôme Euzenat and Pavel Shvaiko. 2007. *Ontology Matching.* Springer, Secaucus, NJ, USA.
[7] Fellipe Freire, Crishane Freire, and Damires Souza. 2017. Enhancing JSON to RDF Data Conversion with Entity Type Recognition. In *Proceedings of the 13th International Conference on Web Information Systems and Technologies, WEBIST, Porto, Portugal.* SciTe Press, Setúbal, Portugal, 97–106.
[8] Markus Lanthaler and Christian Gütl. 2013. Hydra: A Vocabulary for Hypermedia-Driven Web APIs. In *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013) at the 22nd International World Wide Web Conference (CEUR Workshop Proceedings)*, Christian Bizer, Tom Heath, Tim Berners-Lee, Michael Hausenblas, and Sören Auer (Eds.), Vol. 996. CEUR-WS.org, Rio de Janeiro, Brazil.
[9] Markus Lanthaler, Manu Sporny, and Gregg Kellogg. 2014. *JSON-LD 1.0.* W3C Recommendation. W3C. http://www.w3.org/TR/2014/REC-json-ld-20140116/.
[10] 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. *International Journal of Web and Grid Services* 12, 1 (2016), 26–41.
[11] C. Matthew MacKenzie, Ken Laskey, Francis McCabe, Peter F Brown, and Rebekah Metz. 2006. *Reference Model for Service Oriented Architecture 1.0.* Technical Report. OASIS. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html
[12] Alexander Maedche and Steffen Staab. 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* 16, 2 (March 2001), 8.
[13] Sheila A. McIlraith, Tran C. Son, and Honglei Zeng. 2001. Semantic Web services. *IEEE Intelligent Systems* 16, 2 (March 2001), 46–53.
[14] Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. 2012. *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition).* W3C Recommendation. W3C. http://www.w3.org/TR/2012/REC-owl2-syntax-20121211/
[15] Thi Thanh Sang Nguyen and Haiyan Lu. 2016. Domain Ontology Construction Using Web Usage Data. In *Advances in Artificial Intelligence*. Springer, Cham, Switzerland, 338–344.
[16] Joyce Aline Oliveira and José Jorge Lima Dias Junior. 2016. Uma Visão Tridimensional do Reúso em Arquitetura Orientada a Serviços. In *XII Brazilian Symposium on Information Systems*. SBC, Florianópolis, 409–416.
[17] Sara Salem and Samir AbdelRahman. 2010. A Multiple-domain Ontology Builder. In *Proc. of the 23rd Int. Conf. on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, USA, 967–975.
[18] 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. *International Journal of Web Information Systems* 13 (2017), 302–333. Issue 3.
[19] Fabian 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.
[20] Davide Tosi and Sandro Morasca. 2015. Supporting the semi-automatic semantic annotation of web services: A systematic literature review. *Information and Software Technology* 61 (2015), 16–32.
[21] Luis Villaca and Leonardo Azevedo. 2018. An Event-Oriented Microservice Architecture for the Integration of Frequently Updated Data from Polyglot Databases. In *XI Workshop de Teses e Dissertações em Sistemas de Informação*. SBC, Caxias do Sul, Brasil, 1–4.
[22] Yuangang Yao, Hui Liu, Jin Yi, Haiqiang Chen, Xianghui Zhao, and Xiaoyu Ma. 2014. An automatic semantic extraction method for web data interchange. In *6th Int. Conf. on Computer Science and Inf. Technology*. IEEE, Washington, USA, 148–152.