10-2019

Knowledge Discovery from RDF Data stored in NoSQL databases

Isabel Ferreira

José Luís Pereira

Ana Alice Baptista

Descoberta de Conhecimento em Dados RDF Armazenados em Bases de Dados NoSQL

Knowledge Discovery from RDF Data stored in NoSQL databases

Isabel Ferreira, Universidade do Minho, Centro Algoritmi, Portugal, a72245@alunos.uminho.pt
José Luís Pereira, Universidade do Minho, Centro Algoritmi, Portugal, jlmp@dsi.uminho.pt
Ana Alice Baptista, Universidade do Minho, Centro Algoritmi, Portugal, analice@dsi.uminho.pt

Resumo

Atualmente, a existência de grandes volumes de dados sugere a utilização de ferramentas capazes de os processar e de facilitar o processo de descoberta de novo conhecimento. A descoberta de novos factos, que não estavam explícitos anteriormente, pode ser crucial para os processos de tomada de decisão. Nesse contexto, este artigo apresenta uma revisão da literatura relevante relativa a normas da Web Semântica, repositórios de dados RDF (*Resource Description Framework*) e mecanismos de inferência disponíveis em repositórios RDF. O objetivo principal é o de relatar como é que a inferência pode ser aplicada e derivar novos factos a partir dos dados já existentes. Para esse efeito, são apresentadas inferências obtidas a partir de um conjunto de regras pré-definidas sobre dados de publicações científicas armazenados numa base de dados NoSQL designada de MarkLogic.

Palavras-chave: Repositórios RDF; RDF; SPARQL; Inferência; Bases de Dados NoSQL

Abstract

Currently, the existence of large amounts of data suggests the use of tools capable of processing them and facilitate the process of finding new knowledge. The discovery of new facts that were not previously explicit in data can be crucial to decision-making processes. In this article, we present a survey on Semantic Web standards, stores of RDF data (Resource Description Framework) and inference mechanisms available in RDF stores. The main goal is to report how inference can be applied and derive new facts from existing data. For this purpose, we demonstrate inferences obtained from a set of predefined rules over data about scientific publications stored in a NoSQL database designated of MarkLogic.

Keywords: *RDF stores, RDF, SPARQL, Inference, NoSQL Databases*

1. INTRODUÇÃO

A quantidade de dados com que as organizações têm de lidar tem vindo a aumentar exponencialmente. Assim sendo, torna-se crucial para estas a utilização de mecanismos que lhes permitam o processamento inteligente de dados com o intuito de conseguirem retirar ilações que auxiliem nos seus processos de tomada de decisão. Para isso, é necessária a utilização de ferramentas sofisticadas que permitam o processamento de grandes quantidades de dados. No caso particular

ISSN 2183-489X

deste trabalho, debruçamo-nos sobre dados vocacionados para a maximização de interoperabilidade semântica – dados RDF (*Resource Description Framework*).

As bases de dados NoSQL tem vindo a ser utilizadas em repositórios RDF para o processamento de dados devido à sua capacidade de lidarem com grandes volumes de dados. Para além disso, existem repositórios RDF que assentam em base de dados NoSQL e que possuem mecanismos que facilitam a descoberta de novos conhecimentos a partir dos dados existentes e de informações adicionais (ontologias e conjuntos de regras). Assim sendo, pretende-se explorar as capacidades de processamento inteligente dos repositórios RDF assentes em bases de dados NoSQL relativamente à descoberta de novos factos que não estão explícitos nos dados.

Em termos de estrutura, este artigo começa por fazer, na segunda secção, uma contextualização sobres duas normas da Web Semântica, o RDF e o SPARQL. Posteriormente, na terceira secção, é feita uma caracterização dos repositórios RDF e da sua taxonomia, seguindo-se, na quarta secção, uma contextualização da inferência e das técnicas de raciocínio associadas à mesma. Na quinta secção, são apresentadas inferências obtidas a partir de regras customizadas aplicadas a dados de publicações científicas. Finalmente, na sexta secção, retiram-se algumas conclusões.

2. NORMAS DA *WEB* SEMÂNTICA: RDF E SPARQL

A *World Wide Web*, na altura em que foi concebida, tinha como propósito que o seu conteúdo fosse compreendido pelas pessoas. Posteriormente, de modo a permitir que o seu conteúdo pudesse ser também compreendido e processado pelas máquinas – *machine-readable*, surgiu a ideia da Web Semântica (Berners-Lee, 1994). Nesse contexto, foi desenvolvida de base uma *framework* que permite a descrição e a representação dos recursos existentes na Web, bem como das suas relações: o RDF.

2.1. RDF

Resource Description Framework, também conhecido por RDF, foi proposto pela *World Wide Web* Consortium (W3C) e trata-se da *framework* para a representação de informação sobre recursos na Web (Manola, Miller, & McBride, 2014). Através do RDF é possível explicar como é que duas "coisas" se relacionam através de triplos (Haque & Perkins, 2012). O RDF proporciona a representação de conhecimento sobre um recurso particular utilizando um conjunto de *statements* RDF na forma de triplos que assumem o formato de sujeito-predicado-objeto (s,p,o) (MahmoudiNasab & Sakr, 2012), ou recurso-propriedade-valor. Um triplo pode ser visto como se tratando de um par de entidades que estão ligadas através de uma relação nomeada (*named relationship*) ou então de uma entidade associada com um valor de atributo nomeado (*named* atribute *value*) (Zeng & Zou, 2018). O sujeito de um triplo corresponde ao recurso que se pretende descrever e deve ser identificável. Para além disso, o sujeito pode ser um IRI (*internationalized resource*

identifier) ou um nó vazio (*blank* node). O predicado é um IRI e apresenta uma propriedade, ou seja, a característica a ser descrita de um recurso particular. O objeto é o valor da caraterística descrita do sujeito e pode ser um IRI, um nó vazio ou um literal (Cyganiak et al., 2014).

Um conjunto de triplos é designado de grafo RDF (Cyganiak et al., 2014), no qual os sujeitos e objetos são denominados de vértices ou nós e os predicados por arcos (Zeng & Zou, 2018). Os nós de um grafo RDF podem ser divididos em três categorias distintas: nós *IRIs* (*IRIs* nodes), nós literais (*literals* nodes) ou nós brancos ou vazios (*blank* nodes) (Cyganiak et al., 2014). Uma vez que os arcos têm apenas uma direção (do sujeito para o predicado), este permitem caraterizar um grafo RDF como sendo um grafo orientado (Pan, Zhu, Liu, & Ning, 2018). Na Figura 1 é possível observar um grafo RDF com três nós (um sujeito e dois objetos) e dois arcos a fazer a ligação entre os nós (predicados "born" e "playsFor").

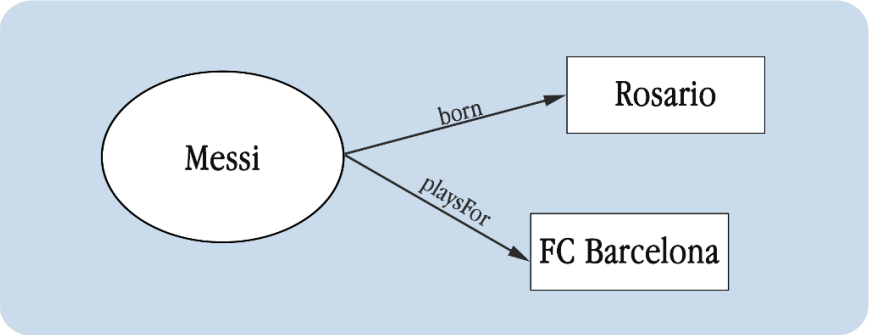

Figura 1 - Grafo RDF orientado. Adaptado de (Chawla, Singh, Pilli, & Govil, 2016).

O RDF tem vindo a ser muito utilizado pelas organizações para o armazenamento de dados devido à grande quantidade de dados (*Big Data*) e pela simplicidade e flexibilidade do modelo RDF (Wu, Zhou, Yuan, Liu, & Jin, 2015). Para além disso, o RDF permite que a informação seja trocada entre diferentes aplicações sem que o significado da mesma seja perdido (Manola et al., 2014).

2.2. SPARQL

SPARQL, abreviação de SPARQL *Protocol and* RDF *Query Language*, é uma linguagem de consulta recomendada pela W3C para RDF e que é semelhante ao SQL (Lee & Liu, 2013). Esta linguagem de consulta é utilizada para expressar consultas a várias fontes de dados, desde que os dados estejam em RDF (Seaborne & Harris, 2013). Esta linguagem sofreu em 2013 uma atualização, na qual o SPARQL 1.1 trata-se de "um conjunto de especificações que fornece linguagem e protocolos para consultar e manipular grafos RDF encontrados na Web ou em repositórios RDF" (The W3C SPARQL Working Group, 2013).

As consultas SPARQL consistem em padrões de triplos, em que cada um destes pode ser composto por variáveis no sujeito, predicado ou objeto (Chen, Trouve, Murakami, & Fukuda, 2017). As variáveis são assinaladas através de pontos de interrogação ("?") e podem ser colocadas em qualquer

parte de um triplo, dando assim origem aos padrões de triplos representados na Figura 2. Portanto, quando uma consulta SPARQL contêm um conjunto de padrões de triplos, esse conjunto é designado de basic *graph pattern* (BGP) (Prud'hommeaux & Seaborne, 2008).

Os resultados da consulta advêm da correspondência do basic *graph pattern* com um grafo RDF (Schätzle, Przyjaciel-Zablocki, Skilevic, & Lausen, 2015). Ao efetuar esta correspondência, está a ser realizado o processamento da consulta SPARQL no qual o resultado obtido é um conjunto de subgrafos do grafo RDF que combinam com os padrões de triplos definidos na consulta SPARQL (Lee & Liu, 2013).

1	(s p o)
2	(?s ?p ?o)
3	(?s p o)
4	(s ?p o)
5	(s p ?o)
6	(?s ?p o)
7	(s ?p o)
8	(?s p ?o)

Figura 2 - Padrões possíveis de triplos RDF. Retirado de (Chawla et al., 2016).

Na Figura 3 é possível observar uma consulta SPARQL (a), um grafo RDF (b) e a correspondência obtida da combinação da consulta SPARQL com o grafo RDF (c).

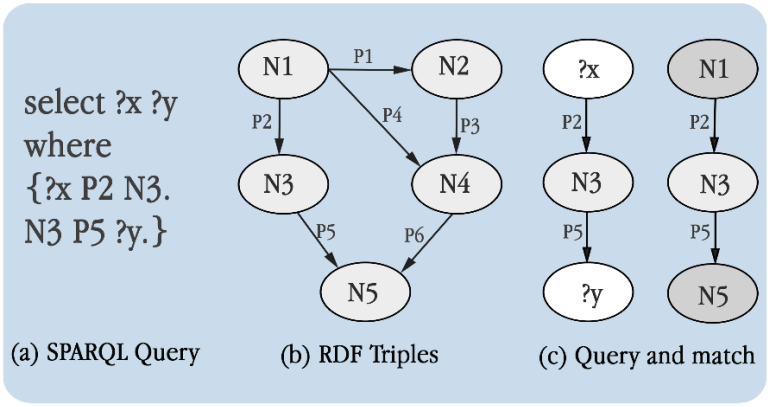

Figura 3 - Representação de uma consulta SPARQL, de triplos RDF e da correspondência da consulta SPARQL. Retirado de (Chen et al., 2017).

Devido à sua semelhança com a linguagem SQL, as consultas SPARQL normalmente seguem o formato de "SELECT a FROM b WHERE c OPTIONAL d" (Haque & Perkins, 2012). Para além da cláusula SELECT, existem outras três cláusulas de retorno que podem ser utilizadas em alternativa e que são o CONSTRUCT, o ASK e o DESCRIBE. A escolha de qual destas cláusulas deve ser empregue na consulta depende do resultado que se pretende obter (Pan et al., 2018).

3. REPOSITÓRIOS RDF

Um repositório RDF trata-se de uma base de dados que é concebida propositadamente com a finalidade de permitir o armazenamento de dados RDF bem como a posterior recuperação dos mesmos (Modoni, Sacco, & Terkaj, 2014) através de consultas aos dados RDF (Sankar, Sayed, & Bani-Younis, 2014).

Os repositórios RDF são compostos por duas componentes principais, nomeadamente, o repositório e o *middleware* (Iancu & Georgescu, 2018; Modoni et al., 2014). O repositório corresponde ao sistema de ficheiros ou à base de dados responsável por efetuar o armazenamento e a gestão dos triplos, enquanto que o *middleware* é o responsável por facilitar a comunicação com o repositório uma vez que ambos estão em comunicação constante (Modoni et al., 2014) (ver Figura 4).

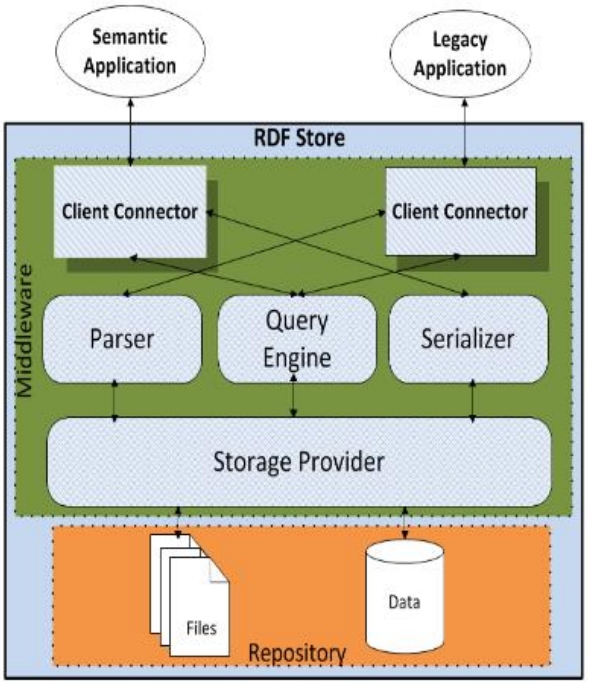

Figura 4 - Exemplo de uma arquitetura de um repositório RDF. Retirado de (Modoni et al., 2014).

Iancu e Georgescu (2018) e Modoni et al. (2014), apresentam as características técnicas mais relevantes na escolha de um repositório RDF. Em ambos os casos são referidas as características em termos de licenças, sistema operativo, linguagens de programação e normas da Web Semântica. Assim sendo, na seleção de um repositório RDF devem ser considerados os seguintes aspetos:

- Normas de Web Semântica que são suportadas: os requisitos mínimos são o RDF, OWL (Web *Ontology Language*) e SPARQL (Iancu & Georgescu, 2018);
- Linguagens de programação suportadas: quais são as linguagens de programação que permitem que o repositório RDF seja conectado com outras aplicações. Isto pode ser útil na vertente do acesso e da gestão dos dados armazenados de forma local ou então a partir de uma aplicação ou cliente (Iancu & Georgescu, 2018). Com esta característica é possível

julgar a capacidade de um repositório RDF interagir com outra aplicação (Modoni et al., 2014);

- Suporte para raciocínio: permite que novo conhecimento seja adquirido através de inferências lógicas. Esta característica trata-se de uma componente importante na Web Semântica (Iancu & Georgescu, 2018);
- Tipo de licença: tem em consideração duas categorias que são os repositório RDF comerciais e os *open source* (Modoni et al., 2014). Os repositórios RDF *open source* oferecem vantagens em termos de flexibilidade e de baixos custos (ou nenhuns), no entanto os outros oferecem uma maior probabilidade de ter um bom suporte (Iancu & Georgescu, 2018);
- Data de lançamento da última versão: é um característica importante porque se a data do lançamento da última versão for antiga, então existe uma grande probabilidade de o repositório RDF não ser suportado ou de não ser possível, devido à incompatibilidade de versões, trabalhar com outras soluções (Iancu & Georgescu, 2018);
- Sistema operativo: diz respeito aos sistemas operativos nos quais o repositório RDF pode ser instalado (Modoni et al., 2014), ou seja, com os quais é compatível. Esta característica é considerada em Iancu e Georgescu (2018) como sendo a menos relevante.

Tendo em conta as características mencionadas acima, foram consideradas no top de repositórios RDF o AlegroGraph, GraphDB, MarkLogic, Mulgara, Profium Sense, RDF4, Stardog, Apache Jena e o Oracle Database 12c (Iancu & Georgescu, 2018).

Os repositórios RDF podem ser classificados de acordo com duas vertentes a nível do armazenamento dos dados: a lógica e a física (Ma, Capretz, & Yan, 2016). No que diz respeito ao armazenamento lógico, os repositórios RDF podem ser categorizados como sendo nativos e não nativos, de acordo com as estratégias de armazenamento que os mesmos adotam e a sua compatibilidade com o modelo RDF (Faye, Curé, & Blin, 2012).

Quando se fala em repositórios RDF nativos, refere-se a sistemas que foram construídos de raiz e que utilizam o modelo de dados RDF para consultar e armazenar (Pan et al., 2018) de forma permanente os dados RDF no sistema de ficheiros (Sankar et al., 2014). Os repositórios RDF nativos podem ainda ser classificados como sendo baseados em memória ou em disco (Faye et al., 2012).

Quanto aos repositórios RDF não nativos, estes dizem respeito a sistemas que são implementados através da incorporação de uma camada RDF em cima de bases de dados já existentes (Pan et al., 2018), com o intuito de armazenar os dados RDF de forma permanente (Faye et al., 2012). Atualmente, existem inúmeras soluções de repositórios RDF suportados em bases de dados relacionais (Pan et al., 2018). Assim sendo, os repositórios RDF podem ser classificados como RDBMS quando são implementados sobre bases de dados relacionais e como NoSQL quando

assentam sobre uma base de dados NoSQL (Hassan & Bansal, 2018). No caso dos repositórios RDF que assentam numa base de dados relacional (RDBMS), estes podem ser divididos, por sua vez, nos que possuem esquema (*schema-based*) e nos que estão livres de esquema (*schema-free*) (Pan et al., 2018), uma vez que para este tipo de repositórios RDF é necessário selecionar o esquema mais apropriado de acordo com a forma como se pretende distribuir os triplos RDF (Banane, Belangour, & El Houssine, 2019). Por sua vez, os repositórios RDF suportados em bases de dados NoSQL dividem-se em quatro categorias, tendo em conta os modelos existentes destas bases de dados: as bases de dados chave-valor; as bases de dados orientadas a colunas; bases de dados orientadas a grafos; e as bases de dados orientadas a documentos (Banane et al., 2019; Pan et al., 2018).

Embora exista um maior número de repositórios RDF suportados em bases de dados relacionais (Pan et al., 2018), estas não são as mais adequadas dada a natureza dinâmica dos dados RDF, ou seja, visto que o esquema dos dados nem sempre é conhecido (Haque & Perkins, 2012). Para além disso, nos últimos anos o volume de dados RDF tem aumentado significativamente (*Big Data*), dificultando assim o seu processamento em bases de dados relacionais. Nesse sentido, têm vindo a ser utilizadas bases de dados NoSQL para colmatar esta limitação (Pan et al., 2018).

Na Figura 5, apresenta-se uma perspetiva das várias classificações relativas a repositórios RDF.

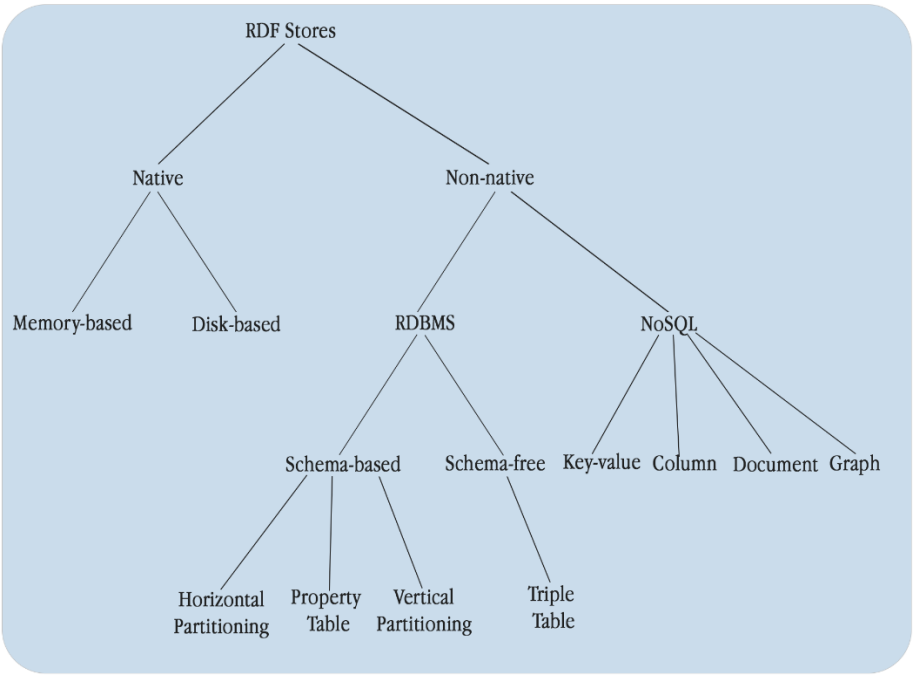

Figura 5 - Taxonomia dos repositórios RDF. Adaptado de (Pan et al., 2018).

4. INFERÊNCIA

A existência de mecanismos para a Web Semântica que proporcionem o processamento inteligente dos dados é uma necessidade para determinadas aplicações (Rattanasawad, Buranarach, Saikaew, & Supnithi, 2018). Por conseguinte, surge então o termo "inferência" como mecanismo que permite a

descoberta de novos relacionamentos com base nos dados existentes e com base em informação adicional (W3C, n.d.), ou seja, que facilita a descoberta de padrões que não estão automaticamente visíveis nos dados.

```
<LinkedDataBook>   <rdf:type>   <Book>
<LinkedDataStory>   <rdf:type>   <Article>
<Book>   <rdfs:subClassOf>   <Publication>
<Article>   <rdfs:subClassOf>   <Publication>
```

Figura 6 - Exemplo de um conjunto de triplos. Retirado de (Sakr, Wylot, Mutharaju, Le Phuoc, & Fundulaki, 2018).

Por exemplo, na Figura 6, facilmente se conclui, através da observação dos triplos, que tanto o <LinkedDataBook> como o <LinkedDataStory> são uma <Publication>. No entanto, para uma máquina extrair estas conclusões lógicas necessita de um "raciocinador" (Sakr et al., 2018).

O "raciocinador" (*reasoner*), conhecido também como motor de inferência (*inference engine*) (Rattanasawad et al., 2018; Singh & Karwayun, 2010), trata-se de um software que permite que sejam derivados novos factos, ou seja, derivados novos triplos RDF a partir da informação já existente (Rattanasawad et al., 2018; Rattanasawad, Saikaew, Buranarach, & Supnithi, 2013; Singh & Karwayun, 2010). É fundamental, para isso, representar as inferências lógicas que fornecem suporte ao raciocínio automatizado (Khamparia & Pandey, 2017) e que, além disso, podem ser empregues para dar resposta a questões particulares sobre os dados (Sakr et al., 2018).

Para além da informação existente, também podem ser derivados novos factos a partir de informação adicional proveniente de ontologias (Beckett & Berners-Lee, 2008; Shi, Chong, & Yan, 2018) e de conjuntos de regras (Rattanasawad et al., 2013; W3C, n.d.). Uma ontologia trata-se de uma "*especificação formal e explícita de uma conceptualização compartilhada*" (Studer, Benjamins, & Fensel, 1998) que permite a inferência de novos relacionamentos quando aplicada em conjunto com um motor de inferência (Shi et al., 2018). Por outro lado, uma regra permite através de condições efetuar a representação de conhecimento e é definida na forma de cláusulas *If-Then*, nas quais o *If* é o antecedente (contêm as condições) e o *Then* é o consequente (contêm as conclusões). Deste modo, são derivados novos factos quando os motores de inferência baseados em regras conseguem obter combinações entre as condições definidas nas regras e os dados existentes (Rattanasawad et al., 2013).

No que toca às linguagens em que as regras podem ser expressas, os motores de inferência podem suportar as suas próprias linguagens de regras ou então as normas existentes para linguagens de regras (Rattanasawad et al., 2018).

Existem duas estratégias de raciocínio para os motores de inferência e que são o *forward chaining* e o *backward chaining* (Khamparia & Pandey, 2017; Rattanasawad et al., 2018; Sakr et al., 2018;

Singh & Karwayun, 2010) e que permitem decidir que regra deve ser executada (Ting, Kadir, Sembok, Ahmad, & Azman, 2014).

Em relação ao *forward chaining*, esta estratégia inicia-se a partir dos factos/dados já existentes e utiliza regras por forma a derivar outros factos/dados, ou seja, inferir todos os factos/dados possíveis (Rattanasawad et al., 2018; Singh & Karwayun, 2010). Por outro lado, o *backward chaining* inicia-se a partir de uma lista de metas, isto é, de uma conclusão (Rattanasawad et al., 2018) ou de uma hipótese (Singh & Karwayun, 2010) e retrocede com o intuito de encontrar todas as possíveis soluções (Predoiu & Grimm, 2005; Rattanasawad et al., 2018), ou seja, os triplos RDF que suportem essa conclusão. Em suma, a estratégia do *backward chaining* ocorre no momento das consultas dos dados enquanto que no caso do *forward chaining* ocorre antecipadamente (Urbani, Kotoulas, Maassen, Van Harmelen, & Bal, 2012).

5. RESULTADOS DE INFERÊNCIAS

Para a concretização de inferências, foi escolhido o MarkLogic como repositório RDF com o intuito de demonstrar o poder da inferência em termos de processamento inteligente e de descoberta automática de conhecimento, hoje disponível em algumas das novas bases de dados NoSQL. A escolha desta ferramenta ficou a dever-se ao facto de ser o repositório RDF atualmente mais popular (DBEngine, 2019).

Relativamente aos dados, foi selecionada uma amostra de um conjunto de dados em RDF com informação sobre publicações científicas e outras entidades que se relacionam com as mesmas, nomeadamente, autores, áreas de estudo/tópicos e instituições. Este conjunto de dados foram fornecidos pelo *Microsoft Academic Knowledge Graph* (Färber, 2018). Os dados utilizados contêm quatro entidades, como se pode constatar na Figura 7.

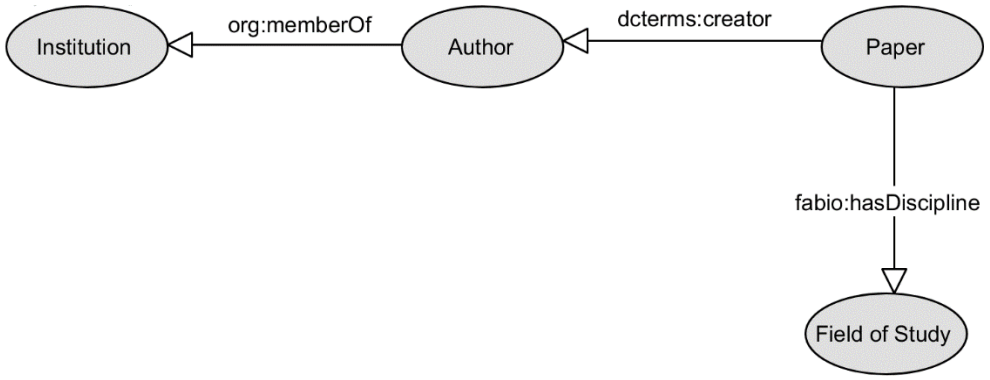

Figura 7 - Representação em grafo RDF dos relacionamentos entre as diferentes entidades.

Tendo em conta as características destes dados, verifica-se que existem determinados factos que não estão explícitos nos dados, mas que podem ser inferidos de forma automática. Nesse sentido, foi então definida uma lista de três metas para as quais se pretende obter resultados e que são: a) quais

são as áreas de investigação de um dado autor, (b) com quem um autor já foi coautor de alguma publicação e (c) quais são os seus colegas de trabalho de um autor. Para obter resultados para estas metas foram então definidas e adicionadas à base de dados três regras customizadas como sendo *default rulesets*. Isto permite que no momento da consulta dos dados seja possível obter os resultados dos triplos inferidos a partir das regras customizadas (*backward chaining*).

Na Figura 8 é possível observar a sintaxe da regra para as áreas de investigação de um autor, bem como a esquematização do triplo inferido através da regra. Estas regras customizadas foram escritas numa linguagem específica do MarkLogic que possui uma sintaxe similar à da cláusula CONSTRUCT das consultas SPARQL (MarkLogic, n.d.).

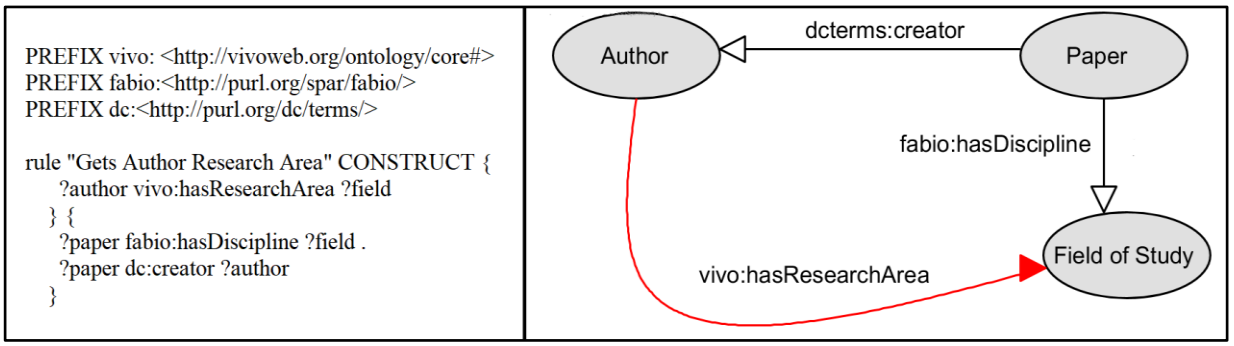

Figura 8 – Regra para as áreas de investigação de um autor e representação do triplo inferido.

Abaixo, a Figura 9 mostra a consulta SPARQL utilizada para apresentar as áreas de investigação do autor identificado por <http://ma-graph.org/entity/196256982> (Figura 10 mostra o resultado).

```
PREFIX foaf:<http://xlmns.com/foaf/0.1/>
PREFIX sioc:<http://rdfs.org/sioc/ns#>
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX ti:<http://purl.org/NET/c4dm/timeline.owl#>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX frbr:<http://purl.org/vocab/frbr/core#>
PREFIX fabio:<http://purl.org/spar/fabio/>
PREFIX cito:<http://purl.org/spar/cito/>
PREFIX datacite:<http://purl.org/spar/datacite/>
PREFIX prism:<http://prismstandard.com/namespaces/1.2/basic/>
PREFIX c4o:<http://purl.org/spar/c4o/>
PREFIX org:<http://www.w3.org/ns/org#>
prefix vivo:<http://vivoweb.org/ontology/core#>
prefix opus:<http://lsdis.cs.uga.edu/projects/semdis/opus#>
prefix sor:   <http://purl.org/net/soron/>

SELECT ?field
  WHERE {<http://ma-graph.org/entity/196256982> vivo:hasResearchArea ?field}
```

Figura 9 – Consulta SPARQL para a visualização das áreas de investigação do autor identificado por <http://ma-graph.org/entity/196256982>.

field
<http://ma-graph.org/entity/105795698>
<http://ma-graph.org/entity/107673813>
<http://ma-graph.org/entity/111350023>
<http://ma-graph.org/entity/11413529>
<http://ma-graph.org/entity/205147927>
<http://ma-graph.org/entity/33923547>
<http://ma-graph.org/entity/57830394>
<http://ma-graph.org/entity/57869625>
<http://ma-graph.org/entity/62100291>
<http://ma-graph.org/entity/67257552>
<http://ma-graph.org/entity/98763669>

Figura 10 - Resultado obtido após a execução da consulta apresentada na Figura 9: áreas de investigação do autor identificado por <http://ma-graph.org/entity/196256982>.

Quanto aos coautores de um autor, na Figura 11 é apresentada a sintaxe da regra das coautorias e a esquematização do triplo inferido a partir da mesma.

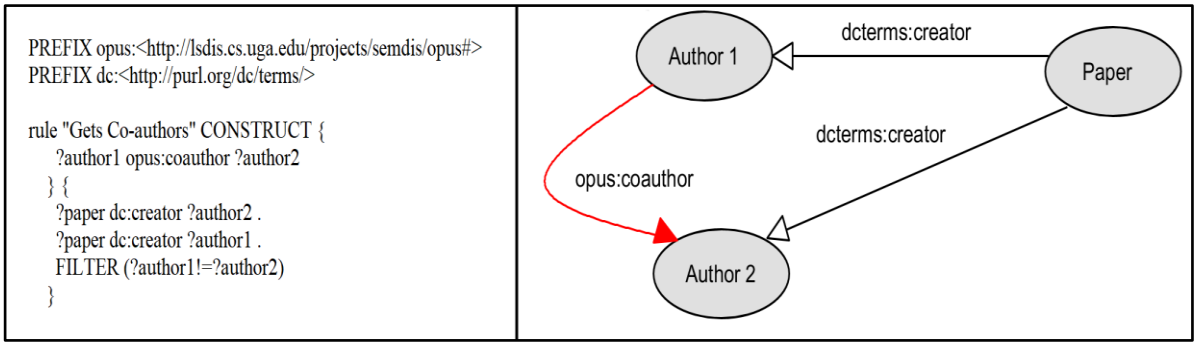

Figura 11 – Regra para as coautorias de um autor e representação do triplo inferido.

A Figura 12 apresenta a consulta SPARQL das coautorias para o autor identificado por <http://ma-graph.org/entity/2405514234>, sendo o resultado obtido desta consulta apresentado na Figura 13.

```
PREFIX foaf:<http://xlmns.com/foaf/0.1/>
PREFIX sioc:<http://rdfs.org/sioc/ns#>
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX ti:<http://purl.org/NET/c4dm/timeline.owl#>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX frbr:<http://purl.org/vocab/frbr/core#>
PREFIX fabio:<http://purl.org/spar/fabio/>
PREFIX cito:<http://purl.org/spar/cito/>
PREFIX datacite:<http://purl.org/spar/datacite/>
PREFIX prism:<http://prismstandard.com/namespaces/1.2/basic/>
PREFIX c4o:<http://purl.org/spar/c4o/>
PREFIX org:<http://www.w3.org/ns/org#>
prefix vivo:<http://vivoweb.org/ontology/core#>
prefix opus:<http://lsdis.cs.uga.edu/projects/semdis/opus#>
prefix sor:   <http://purl.org/net/soron/>

SELECT ?author2
  WHERE {<http://ma-graph.org/entity/2405514234> opus:coauthor ?author2}
```

Figura 12 - Consulta SPARQL para a visualização das coautorias do autor identificado por <http://ma-graph.org/entity/2405514234>.

author2
<http://ma-graph.org/entity/2640195073>
<http://ma-graph.org/entity/2647181028>

Figura 13 - Resultado obtido após a execução da consulta apresentada na Figura 12: coautorias do autor identificado por <http://ma-graph.org/entity/2405514234>.

Quanto aos colegas de trabalho de um dado autor, na Figura 14 encontra-se representada a sintaxe da regra e a esquematização do triplo inferido através da mesma.

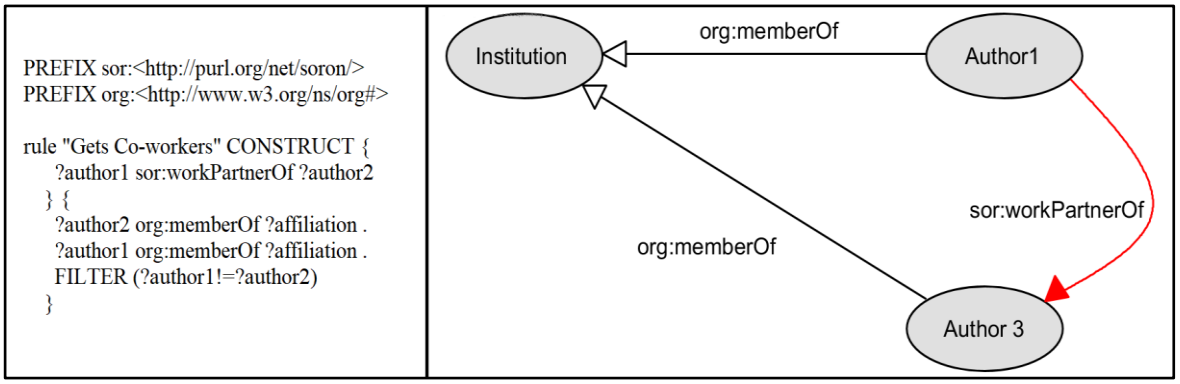

Figura 14 - Regra para as colegas de trabalho de um autor e representação do triplo inferido.

A Figura 15 apresenta a consulta SPARQL dos colegas de trabalho para o autor identificado por <http://ma-graph.org/entity/2481555596>. Na Figura 16 encontram-se representados os resultados obtidos da consulta SPARQL.

```
PREFIX foaf:<http://xlmns.com/foaf/0.1/>
PREFIX sioc:<http://rdfs.org/sioc/ns#>
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX ti:<http://purl.org/NET/c4dm/timeline.owl#>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX frbr:<http://purl.org/vocab/frbr/core#>
PREFIX fabio:<http://purl.org/spar/fabio/>
PREFIX cito:<http://purl.org/spar/cito/>
PREFIX datacite:<http://purl.org/spar/datacite/>
PREFIX prism:<http://prismstandard.com/namespaces/1.2/basic/>
PREFIX c4o:<http://purl.org/spar/c4o/>
PREFIX org:<http://www.w3.org/ns/org#>
prefix vivo:<http://vivoweb.org/ontology/core#>
prefix opus:<http://lsdis.cs.uga.edu/projects/semdis/opus#>
prefix sor:   <http://purl.org/net/soron/>

SELECT ?author2
  WHERE {<http://ma-graph.org/entity/2481555596> sor:workPartnerOf ?author2}
```

Figura 15 - Consulta SPARQL para a visualização dos colegas de trabalho do autor identificado por <http://ma-graph.org/entity/2481555596>.

author2
<http://ma-graph.org/entity/2481510026>

Figura 16 - Resultado obtido após a execução da consulta apresentada na Figura 15: colegas de trabalho do autor identificado por <http://ma-graph.org/entity/2481555596>.

Considerando os resultados apresentados em cima, constata-se que a inferência traz vantagens em termos de processamento inteligente uma vez que, no caso de regras pré-definidas, é capaz de inferir automaticamente quais são os dados que têm as condições necessárias para satisfazer as conclusões definidas nas regras.

A Figura 17 mostra os triplos RDF que são necessários combinar para obter os mesmos resultados de áreas de investigação do autor identificado por <http://ma-graph.org/entity/196256982> que foram apresentados na Figura 9. Assim, é possível constatar que com a inferência é possível reduzir a complexidade das consultas SPARQL que envolvam entidades que não possuam uma ligação direta.

```
PREFIX rdfs:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX mag:<http://ma-graph.org/>
PREFIX foaf:<http://xlmns.com/foaf/0.1/>
PREFIX sioc:<http://rdfs.org/sioc/ns#>
PREFIX dc:<http://purl.org/dc/terms/>
PREFIX ti:<http://purl.org/NET/c4dm/timeline.owl#>
PREFIX dbo:<http://dbpedia.org/ontology/>
PREFIX frbr:<http://purl.org/vocab/frbr/core#>
PREFIX fabio:<http://purl.org/spar/fabio/>
PREFIX cito:<http://purl.org/spar/cito/>
PREFIX datacite:<http://purl.org/spar/datacite/>
PREFIX prism:<http://prismstandard.com/namespaces/1.2/basic/>
PREFIX c4o:<http://purl.org/spar/c4o/>
PREFIX org:<http://www.w3.org/ns/org#>
prefix vivo:<http://vivoweb.org/ontology/core#>
prefix opus:<http://lsdis.cs.uga.edu/projects/semdis/opus#>
prefix sor:   <http://purl.org/net/soron/>

SELECT ?field
  WHERE {?paper fabio:hasDiscipline ?field .
         ?paper dc:creator <http://ma-graph.org/entity/196256982>}
```

Figura 17 – Consulta SPARQL para obter as áreas de investigação do autor identificado por <http://ma-graph.org/entity/196256982> sem recurso à inferência.

6. CONCLUSÃO

Enormes quantidades de dados são produzidas diariamente e, portanto, torna-se essencial que as organizações consigam tirar o melhor partido deste recurso de forma a adquirirem novo conhecimento para melhor suportar os seus processos de decisão. Neste âmbito, o RDF ao permitir que os dados sejam representados sob a forma de triplos, possibilita que as organizações retirem benefícios da sua flexibilidade e usufruam do seu foco nos relacionamentos entre os dados. Para além disso, vários repositórios RDF assentes em bases de dados NoSQL possuem mecanismos que facilitam o processo de inferência de novo conhecimento.

A inferência promove a descoberta de relacionamentos que não estavam inicialmente explícitos entre os dados, ou seja, novos factos. Por isso, através do processamento inteligente dos dados é proporcionada a integração destes visto que, através da inferência é possível derivar novos factos que envolvem entidades sem ligação direta entre elas. Para além disso, também se constatou que, através das regras pré-definidas para inferir novos factos, é reduzida a complexidade de consultas

SPARQL que implicam a combinação de vários triplos RDF. Através das regras é possível inferir novos triplos sempre que os dados possuam as condições necessárias para satisfazer a meta associada a uma dada regra.

O próximo passo envolve a utilização de um maior volume de dados e a aplicação de novos conjuntos de regras de forma a descobrir padrões interessantes nos dados tendo em conta a riqueza do seu conteúdo.

AGRADECIMENTOS

Este trabalho é financiado pelo FEDER – Fundo Europeu de Desenvolvimento Regional através do COMPETE 2020 – Programa Operacional Competitividade e Internacionalização (POCI) e por fundos nacionais através da FCT – Fundação para a Ciência e a Tecnologia no âmbito dos Projetos IViSSEM - PTDC/COM-INF/28284/2017 e UID/CEC/00319/2019.

REFERÊNCIAS

Banane, M., Belangour, A., & El Houssine, L. (2019). Storing RDF Data into Big Data NoSQL Databases. In *Advances in Intelligent Systems and Computing* (Vol. 756, pp. 69–78). https://doi.org/10.1007/978-3-319-91337-7_7

Beckett, D., & Berners-Lee, T. (2008). Turtle - Terse RDF Triple Language. Retrieved from https://www.w3.org/TeamSubmission/2008/SUBM-turtle-20080114/#sec-intro

Berners-Lee, T. (1994). W3 future directions. Retrieved November 18, 2018, from https://www.w3.org/Talks/WWW94Tim/

Chawla, T., Singh, G., Pilli, E. S., & Govil, M. C. (2016). Research issues in RDF management systems. In *2016 International Conference on Emerging Trends in Communication Technologies (ETCT)* (pp. 1–5). IEEE. https://doi.org/10.1109/ETCT.2016.7882968

Chen, H., Trouve, A., Murakami, K. J., & Fukuda, A. (2017). Semantic image retrieval for complex queries using a knowledge parser. *Multimedia Tools and Applications*, pp. 1–19. https://doi.org/10.1007/s11042-017-4932-2

Cyganiak, R., Wood, D., Lanthaler, M., Graham, K., Carrol, J. J., & McBride, B. (2014). RDF 1.1 Concepts and Abstract Syntax. Retrieved June 17, 2019, from https://www.w3.org/TR/rdf11-concepts/#dfn-iri

DBEngine. (2019). DB-Engines Ranking of RDF Stores. Retrieved May 1, 2019, from https://db-engines.com/en/ranking

Färber, M. (2018). The Microsoft Academic Graph in RDF: A Linked Data Source with 8 Billion Triples of Scholarly Data. https://doi.org/10.5281/ZENODO.2159723

Faye, D. C., Curé, O., & Blin, G. (2012). *A survey of RDF storage approaches*. *Informatique et Mathématiques Appliquées* (Vol. 15). Retrieved from https://hal.inria.fr/hal-01299496

Haque, A., & Perkins, L. (2012). Distributed RDF Triple Store Using HBase and Hive. *University of Texas at Austin*.

Hassan, M., & Bansal, S. K. (2018). Semantic Data Querying over NoSQL Databases with Apache Spark. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)* (pp. 364–371). IEEE. https://doi.org/10.1109/IRI.2018.00061

Iancu, B., & Georgescu, T. M. (2018). Saving Large Semantic Data in Cloud: A Survey of the Main DBaaS Solutions. *Informatica Economica*, *22*(1/2018), 5–16. https://doi.org/10.12948/issn14531305/22.1.2018.01

Khamparia, A., & Pandey, B. (2017). Comprehensive analysis of semantic web reasoners and tools: a survey. *Education and Information Technologies*, *22*(6), 3121–3145. https://doi.org/10.1007/s10639-017-9574-5

Lee, K., & Liu, L. (2013). Scaling queries over big RDF graphs with semantic hash partitioning. *Proceedings of the VLDB Endowment*, *6*(14), 1894–1905. https://doi.org/10.14778/2556549.2556571

Ma, Z., Capretz, M. A. M., & Yan, L. (2016). Storing massive Resource Description Framework (RDF) data: A survey. *Knowledge Engineering Review*. https://doi.org/10.1017/S0269888916000217

MahmoudiNasab, H., & Sakr, S. (2012). AdaptRDF: adaptive storage management for RDF databases. *International Journal of Web Information Systems*, *8*(2), 234–250. https://doi.org/10.1108/17440081211241978

Manola, F., Miller, E., & McBride, B. (2014). RDF 1.1 Primer. Retrieved June 17, 2019, from https://www.w3.org/TR/rdf11-primer/

MarkLogic. (n.d.). Inference (Semantic Graph Developer's Guide) — MarkLogic 9 Product Documentation. Retrieved May 20, 2019, from https://docs.marklogic.com/guide/semantics/inferencing

Modoni, G. E., Sacco, M., & Terkaj, W. (2014). A survey of RDF store solutions. In *2014 International Conference on Engineering, Technology and Innovation: Engineering Responsible Innovation in Products and Services, ICE 2014*. https://doi.org/10.1109/ICE.2014.6871541

Pan, Z., Zhu, T., Liu, H., & Ning, H. (2018). A survey of RDF management technologies and benchmark datasets. *Journal of Ambient Intelligence and Humanized Computing*, *9*(5), 1693–1704. https://doi.org/10.1007/s12652-018-0876-2

Predoiu, L., & Grimm, S. (2005). Reasoner technology scan and recommendation. Data, information and process integration with semantic web services, FP6-507483. Retrieved from https://scholar.google.pt/scholar?hl=pt-PT&as_sdt=0%2C5&q=%22Reasoner+technology+scan+and+recommendation.+Data%2C+information+and+process+integration+with+semantic+web+services%22&btnG=

Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL Query Language for RDF. *W3C Recommendation*. Retrieved November 18, 2018, from https://www.w3.org/TR/rdf-sparql-query/

Rattanasawad, T., Buranarach, M., Saikaew, K. R., & Supnithi, T. (2018). A comparative study of rule-based inference engines for the semantic web. In *IEICE Transactions on Information and Systems* (Vol. E101D, pp. 82–89). https://doi.org/10.1587/transinf.2017SWP0004

Rattanasawad, T., Saikaew, K. R., Buranarach, M., & Supnithi, T. (2013). A review and comparison of rule languages and rule-based inference engines for the Semantic Web. In *2013 International Computer Science and Engineering Conference, ICSEC 2013* (pp. 1–6). IEEE. https://doi.org/10.1109/ICSEC.2013.6694743

Sakr, S., Wylot, M., Mutharaju, R., Le Phuoc, D., & Fundulaki, I. (2018). Distributed Reasoning of RDF Data. In *Linked Data* (pp. 109–126). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-73515-3_6

Sankar, S., Sayed, A., & Bani-Younis, J. A. (2014). A Schematic Analysis on Selective-RDF Database Stores. *International Journal of Computer Applications*, *86*(11), 21–28. https://doi.org/10.5120/15030-3348

Schätzle, A., Przyjaciel-Zablocki, M., Skilevic, S., & Lausen, G. (2015). S2RDF: RDF Querying with SPARQL on Spark. https://doi.org/10.14778/2977797.2977806

Seaborne, A., & Harris, S. (2013). SPARQL 1.1 Query Language. Retrieved November 18, 2018, from https://www.w3.org/TR/sparql11-query/

Shi, H., Chong, D., & Yan, G. (2018). Evaluating an optimized backward chaining ontology reasoning system with innovative custom rules. *Information Discovery and Delivery*, *46*(1), 45–56. https://doi.org/10.1108/IDD-10-2017-0070

Singh, S., & Karwayun, R. (2010). A comparative study of inference engines. In *ITNG2010 - 7th International Conference on Information Technology: New Generations* (pp. 53–57). IEEE. https://doi.org/10.1109/ITNG.2010.198

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, *25*(1–2), 161–197. https://doi.org/10.1016/S0169-023X(97)00056-6

The W3C SPARQL Working Group. (2013). SPARQL 1.1 Overview. Retrieved June 17, 2019, from https://www.w3.org/TR/sparql11-overview/

Ting, M., Kadir, R. A., Sembok, T. M. T., Ahmad, F., & Azman, A. (2014). Feasibility Study Concerning the Use of Reasoning Technique in Semantic Reasoners (pp. 371–381). Springer, Cham. https://doi.org/10.1007/978-3-319-12844-3_32

Urbani, J., Kotoulas, S., Maassen, J., Van Harmelen, F., & Bal, H. (2012). WebPIE: A Web-scale Parallel Inference Engine using MapReduce. *Journal of Web Semantics*, *10*, 59–75. https://doi.org/10.1016/j.websem.2011.05.004

W3C. (n.d.). Inference - W3C. Retrieved April 5, 2019, from https://www.w3.org/standards/semanticweb/inference

Wu, B., Zhou, Y., Yuan, P., Liu, L., & Jin, H. (2015). Scalable SPARQL querying using path partitioning. In *Proceedings - International Conference on Data Engineering* (Vol. 2015-May, pp. 795–806). IEEE. https://doi.org/10.1109/ICDE.2015.7113334

Zeng, L., & Zou, L. (2018, August 9). Redesign of the gStore system. *Frontiers of Computer Science*, pp. 623–641. https://doi.org/10.1007/s11704-018-7212-z