



Modelagem em ontologias de relações de compatibilidade entre produtos em *E-commerce*

Diogo Teles Sant'Anna

Julio Cesar dos Reis

Relatório Técnico - IC-PFG-20-24

Projeto Final de Graduação

2020 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Modelagem em ontologias de relações de compatibilidade entre produtos em *E-commerce*

Diogo Teles Sant’Anna* Julio Cesar dos Reis*

December, 2020

Resumo

Atualmente há um número cada vez maior de vendas em plataformas de *e-commerces*. Clientes com frequência fazem perguntas sobre um produto antes de comprá-lo. As lojas online podem melhorar a experiência do usuário, sua satisfação e também a taxa de vendas ao responder essas perguntas instantaneamente. Muitas perguntas apresentam especial dificuldade para serem respondidas automaticamente por demandarem conhecimento específico. Mais especificamente, há um volume grande de perguntas sobre compatibilidade de produtos no *e-commerce*. Este trabalho objetiva estudar como modelar relações de compatibilidade sobre produtos na construção de um Grafo de Conhecimento (GC) estruturado com base em ontologias. Para esse fim, necessitamos determinar uma ontologia que modele os conceitos envolvidos no domínio e as relações entre eles. Em particular, abordamos possíveis modelagens e comparamos duas ontologias diferentes, exibindo exemplos de sua utilização. A versão final da ontologia é explorada em um GC construído na empresa *GoBots*¹. O GC é uma solução que auxilia a empresa a responder automaticamente questões em linguagem natural em plataformas de *e-commerces*.

1 Introdução

É muito comum clientes fazerem perguntas sobre um produto antes de comprá-lo. Para lojas de *e-commerce* é de grande interesse entender as perguntas feitas pelos consumidores automaticamente via sistemas computacionais. Isso pode ajudar a gerar respostas adequadas e imediatas que não apenas ajudam os compradores como também reduzem custos de uma equipe de pessoas direcionada a responder perguntas manualmente.

*Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil.

¹Empresa líder na América Latina no ramo de inteligência artificial para *e-commerces*

Atualmente, a *GoBots* processa automaticamente perguntas em linguagem natural (LN) visando respondê-las utilizando um sistema baseado em regras previamente definidas manualmente. Mesmo assim, muitas perguntas ainda não podem ser respondidas por falta de conhecimento específico, principalmente perguntas relacionadas à compatibilidade de produtos. Essas são especialmente frequentes e problemáticas no domínio automotivo, correspondendo a cerca de 40% das perguntas feitas. O consumidor faz perguntas como “Esse apoio de braço funciona no palio 2014?”. Para responder a essa pergunta precisa-se de conhecimento específico que codifica se o dado produto é compatível ou não com o carro.

Perguntas em *e-commerce* devem ser respondidas com muita precisão, pois fornecer informações erradas aos clientes pode gerar consequências indesejáveis. Considerando a pergunta dada no exemplo ilustrado, se o sistema erroneamente responde que sim, o cliente pode comprar o produto e eventualmente solicitar devolução por não funcionar como informado. Além disso, se o sistema responde errado afirmando que o produto não serviria, o cliente vai desistir de uma compra que poderia ter feito.

A *GoBots* se empenha em pesquisas e desenvolvimento de soluções para extrair e estruturar conhecimento sobre compatibilidade de produtos. Como trata-se de um conhecimento sobre relações entre produtos e os itens do consumidor, a proposta foi armazenar as informações em um Grafo de Conhecimento (GC). Para se usar um GC, no entanto, deve-se projetar e desenvolver uma ontologia que defina os conceitos a serem representados e as relações entre eles.

Neste trabalho abordamos como elaborar uma ontologia capaz de representar relações de compatibilidade no contexto de *e-commerce*. Exploramos diferentes modelagens possíveis, efetuando análises e demonstrando suas utilizações de maneira comparativa.

O restante deste trabalho está organizado da seguinte forma: A seção 2 apresenta os fundamentos desta pesquisa e também trabalhos relacionados. A seção 3 descreve nossa abordagem para modelar ontologias visando representar relações de compatibilidade, as dificuldades que surgem nessas modelagens e os resultados considerando comparações entre as propostas de modelagem. A seção 4 elabora uma discussão sobre os resultados obtidos e aponta os trabalhos futuros. A seção 5 efetua a conclusão do trabalho.

2 Fundamentos e Trabalhos Relacionados

Propomos modelar uma ontologia apropriada para representar compatibilidades no domínio de *e-commerce*. Uma ontologia \mathcal{O} conceitualiza um domínio em termos de conceitos, atributos e relações [2]. Formalmente, uma ontologia $\mathcal{O} = (\mathcal{C}_{\mathcal{O}}, \mathcal{R}_{\mathcal{O}}, \mathcal{A}_{\mathcal{O}})$ consiste de um conjunto de conceitos $\mathcal{C}_{\mathcal{O}}$ inter-relacionados por relações direcionadas $\mathcal{R}_{\mathcal{O}}$. Cada conceito $c \in \mathcal{C}_{\mathcal{O}}$ tem um identificador único e é associado com um conjunto

de atributos $\mathcal{A}_{\mathcal{O}}(c) = \{a_1, a_2, \dots, a_p\}$. Cada relação $r(c_1, c_2) \in \mathcal{R}_{\mathcal{O}}$ é tipicamente uma tripla (c_1, r, c_2) onde r é uma relação do domínio (*e.g.*, “is_a”, “part_of”, “advised_by”, *etc.*) inter-relacionando c_1 e c_2 .

A ontologia será usada para modelar um GC, e ambos usam *Resource Description Framework* (RDF) como modelo base para descrição computacional do conhecimento. O RDF [4] é um padrão da web semântica oficializado pela W3C e estrutura informações em “triplas”. Essas são compostas sempre por um sujeito, um predicado e um objeto. Como exemplo, tome a seguinte frase em português: “Diogo pesquisa para o IC”. Essa frase tem:

- Um sujeito: *Diogo*
- Um predicado: *pesquisa para*
- Um objeto: *IC*

Por fim, uma tripla RDF que corresponderia à essa frase e ao conhecimento contido nela seria:

`_:Diogo _:pesquisaPara _:IC .`

A *GoBots* obteve resultados na extração e geração de Grafos de Conhecimento utilizados para gerar respostas automáticas. Sant’Anna *et al.* [6] investigaram como conhecimentos de compatibilidade são extraídos de perguntas já respondidas por atendentes humanos. As compatibilidades são armazenadas em um GC que usa a ontologia descrita neste trabalho. Além disso, a investigação citada aborda como o GC é consultado por um serviço desenvolvido e capaz de utilizar os conhecimentos encontrados para formular respostas aos consumidores; e suprir parte da demanda de respostas automáticas para questões envolvendo compatibilidade.

Ontologia é um artefato relevante para estruturar e definir de maneira computacionalmente tratável, e entendemos que ela é útil na modelagem de conhecimento sobre produtos. Fawei *et al.* [1] apresentou uma solução que usa uma ontologia para responder perguntas relacionadas à legislação. O processo de criação da ontologia inclui a revisão de respostas de especialistas para perguntas selecionadas. As respostas então são analisadas sintaticamente, tomando os substantivos, verbos separando os principais e obtendo suas relações. Essas interpretações, no entanto, são feitas com o uso de aplicações como *WordNet* e *StanfordNPL*, que infelizmente ainda não funcionam da mesma forma na Língua Portuguesa, ou estão em estágio inicial.

Lee *et al.* [5] apresentaram uma solução com um sistema que propõe organizar informações sobre produtos de forma que possam ser usados pelo sistema público de compras da Coreia. Autores usaram o sistema como um *Technical Dictionary* (TD) visando não apenas classificar tecnicamente produtos e seus atributos, mas igualmente administrar informações sobre relações entre diferentes produtos.

GCs tem sido usados muito para além do escopo de descrição de produtos ou de respostas automatizadas. Por exemplo, Szekely *et al.* [7] propuseram a criação de GCs para combater tráfico humano a fim de combater dificuldades como explorar dados de fontes heterogêneas. Eles estudaram como escalar a quantidade de dados e filtrar ruídos nos dados. Eles apresentaram uma solução para construir GCs explorando tecnologias semânticas para conciliar os dados continuamente extraído de diferentes fontes. A solução deles abrangeu consultas interativas nos dados. Esses avanços são relevantes por mostrar como escalar um sistema de consulta sobre GCs.

Nossa análise sintética da literatura indica que ontologias e grafos de conhecimentos tem ainda amplo espaço para serem explorados no contexto de e-commerce. Em particular, este trabalho contribui com a análise de soluções para modelagem de uma ontologia para representar compatibilidades.

3 Modelagem da Ontologia

Esta seção apresenta nossas propostas de modelagens da ontologia a ser usada para definir os conceitos e relações necessárias para estruturar compatibilidades. A ontologia é usada para estruturar GCs que codificam informações de compatibilidades de produtos. O restante desta seção está organizado da seguinte forma: a subseção 3.1 define os conceitos chave da ontologia; a subseção 3.2 descreve as abordagens para a sua modelagem.

3.1 Conceitos chave na ontologia

Da demanda de estruturar conhecimentos de compatibilidade, surge a demanda de modelar representações dela e de conceitos em seu contexto. O conceito de compatibilidade se refere a uma relação entre um produto - no caso, um produto vendido na loja online do *e-commerce* -, e um item do consumidor, que o consumidor deseja saber se funciona apropriadamente com o produto. Tome como um exemplo uma loja online vendendo um volante de automóvel. Esse volante é de um fabricante específico e muito provavelmente funciona apenas em determinados modelos de carro. A relação de compatibilidade entre esse volante e um dado modelo de carro é que ditará se o produto funcionará no carro ou não.

O produto deve representar uma instância de produto possível de ser comprado em um *e-commerce*. O produto deve conter um identificador único e idealmente universal, para que possa ser usado para representar um mesmo produto mesmo que em lojas diferentes na plataforma de *e-commerce*. Além de identificadores universais, lojas online também costumam atribuir categorias aos produtos. Retomando o exemplo do volante como um produto, um *e-commerce* poderia classificá-lo na categoria “Acessórios para Carros”.

O conceito de item do consumidor tem o propósito de identificar quaisquer itens, objetos ou posses de um eventual consumidor e que possam ter alguma integração com produtos do *e-commerce*. Um exemplo de item do consumidor é um veículo de posse do cliente. No contexto da Gobots, é comum *e-commerces* venderem peças ou produtos para serem usados em veículos, como rádios, apoios de braços, vidros elétricos, entre outros. O cliente tem um veículo e deseja saber se um desses produtos funcionaria ou caberia adequadamente no seu veículo. Para armazenar essa informação é preciso identificar veículos específicos. A identificação e atributos de um determinado item do consumidor podem ser muito variados e são muito dependentes do seu domínio. No domínio automobilístico aqui retratado, por exemplo, os atributos modelo, marca e ano do veículo normalmente são suficientes para identificar um veículo. Se for necessário representar o domínio de celulares, talvez também seja necessário atributos como tamanho de tela ou processador. A modelagem de compatibilidades também depende, portanto, de um conhecimento acerca do domínio de itens do consumidor que se quer tratar. Mais especificamente, é necessário determinar quais os atributos suficientes para identificar o dado item do consumidor.

Com base nos conceitos de produto e item do consumidor, precisamos representar uma possível compatibilidade entre eles. Note que essa informação não se restringe necessariamente a registrar uma compatibilidade afirmativa. Dado um produto e um item do consumidor, objetivamos registrar a informação de que o produto é de fato compatível, mas igualmente ter a possibilidade de registrar se o produto não é compatível, é compatível sob alguma condição, ou até mesmo se o produto possui compatibilidade universal. São atribuídos à categoria compatibilidade universal os produtos que funcionariam em qualquer item do consumidor dentro de um dado domínio. No domínio automotivo, pode-se citar por exemplo um suporte para celular que se anexa ao vidro e portanto funciona em qualquer carro, independente das especificações.

Essa variabilidade de significados possíveis para uma compatibilidade deu origem ao que denominamos de *tipo de compatibilidade*. Em nossa proposta, uma compatibilidade pode ser dos seguintes tipos:

- Afirmativa: representa a informação de que o produto é de fato compatível;
- Negativa: representa a informação de que o produto não é compatível;
- Condicional: representa que o produto é compatível, dada alguma outra condição;
- Universal: representa que o produto é compatível universalmente, e que deve funcionar em qualquer item do consumidor, dentro do domínio em questão.

```

PREFIX onto: <http://www.example.org/ontology#>
PREFIX graph: <http://www.example.org/knowledgegraph#>

onto:Product rdf:type owl:Class .
onto:ConsumerItem rdf:type owl:Class .
onto:Car rdf:type owl:Class ;
        rdfs:subClassOf onto:ConsumerItem .

```

Código 1: Triplas RDF para representar os conceitos comuns entre ambas abordagens

3.2 Abordagens para a modelagem

Desenvolvemos duas diferentes abordagens para modelar os conceitos de compatibilidade em uma ontologia.

Ambas as abordagens modelam os conceitos de *produto* e *item do consumidor* como classes (cf. Figura 1).

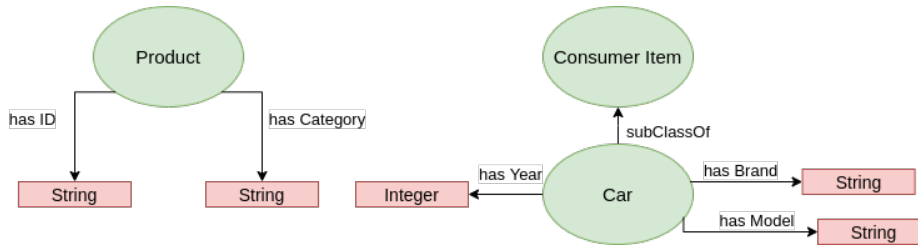


Figura 1: Representação dos conceitos de Produto e Item do Consumidor em uma ontologia.

Produto é definido como uma classe na ontologia, e suas instâncias devem contar com os atributos *ID* e *Category*.

Item do consumidor é definido como uma classe, chamada *Consumer Item*. Ela conta com uma subclasse *Car*, que representa o domínio dos veículos. Se forem utilizados outros tipos de itens do consumidor, representando outros domínios, esses também são modelados como subclasses. Dessa forma, *Consumer Item* se comportará como uma classe abstrata e não deve ter instâncias diretas. As instâncias de *Car* devem contar com os atributos *Year*, *Brand* e *Model*.

O código 1 apresenta a codificação das triplas RDF que representam os conceitos até agora expostos. Essas triplas também são comum entre as duas abordagens que propomos para modelar compatibilidade.

3.2.1 Representando compatibilidade como um predicado

Em uma primeira abordagem, as compatibilidades entre um produto e um item do consumidor são representadas utilizando propriedades de objetos (cf. Figura 2). Essa é a abordagem imediata e mais simples, pois a compatibilidade de fato representa uma relação entre uma instância de produto com uma instância de um item do consumido.

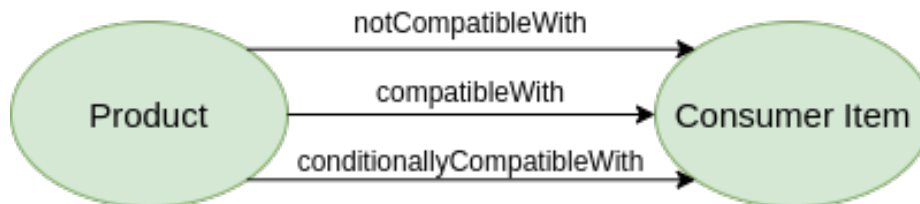


Figura 2: Representação do conceito de compatibilidade como uma propriedade de objeto.

Nessa abordagem, os diferentes tipos de compatibilidades são representados como propriedades de objetos, todas relacionando um produto a um item do consumidor. Assim temos propriedades de nomes *compatibleWith*, *notCompatibleWith* e *conditionallyCompatibleWith*.

O código 2 apresenta um exemplo de triplas RDF representando compatibilidades entre produtos e carros. No exemplo, criamos produtos e carros arbitrários, e dizemos que um *product1* é compatível com um *car1*, e condicionalmente compatível com um *car2*; enquanto um *product2* é compatível com o *car1*, mas não é compatível com o *car2*.

A abordagem apresenta uma solução simples, porém sólida para a representação de compatibilidade desejada. Apresenta, no entanto, algumas desvantagens, como a dificuldade em representar uma compatibilidade universal e a impossibilidade de adicionar metadados às compatibilidades. Os dois problemas serão abordados em detalhes até o fim da subseção.

Nesta abordagem as compatibilidades são sempre representadas como uma relação entre um produto e um item do consumidor específico, porém essa representação não é diretamente aplicável às compatibilidades universais. Quando se diz que um produto é universal, ele passa a ser compatível com qualquer item do consumidor, mas não tem relação com um item específico. Dessa forma, semanticamente não faz sentido representá-la da mesma forma que os outros tipos de compatibilidade. Uma solução alternativa seria criar uma classe *UniversalProduct*, subclasse de *Product*, mas isso fica fora do padrão de representação das compatibilidades.

No ponto de vista prático, é interessante para a GoBots ser capaz de armazenar dados relativos à compatibilidade em si, como a data em que o conhecimento da compatibilidade foi encontrada, ou de onde foi extraída. A abordagem aqui sugerida

```

onto:compatibleWith rdf:type owl:ObjectProperty .
onto:notCompatibleWith rdf:type owl:ObjectProperty .
onto:conditionallyCompatibleWith rdf:type owl:ObjectProperty .

graph:product1 rdf:type onto:Product .
graph:product2 rdf:type onto:Product .
graph:car1 rdf:type onto:Car .
graph:car2 rdf:type onto:Car .

graph:product1 onto:compatibleWith graph:car1 .
graph:product1 onto:conditionallyCompatibleWith graph:car2 .
graph:product2 onto:compatibleWith graph:car1 .
graph:product2 onto:notCompatibleWith graph:car2 .

```

Código 2: Exemplo de triplas RDF para o caso de usarmos compatibilidades como propriedades de objeto. No exemplo, *product1* é compatível com um *car1*, e condicionalmente compatível com um *car2*; e *product2* é compatível com o *car1*, mas não é compatível com o *car2*.

não permite armazenar esse tipo de informação, pois a relação de compatibilidade entre um produto e um item do consumidor é registrada utilizando uma propriedade relacionando duas instâncias. Em RDF não é possível adicionar informações a uma relação como essa.

3.2.2 Representando compatibilidade como uma classe

Como forma de resolver os problemas levantados, desenvolvemos uma nova representação alternativa para a compatibilidade. A representação aborda a possibilidade de representar compatibilidade utilizando uma instância de compatibilidade intermediária entre o produto e o item do consumidor.

É criada uma nova classe na Ontologia para representar o conceito de Compatibilidade. Criamos igualmente uma subclasse de Compatibilidade para representar cada um dos tipos de compatibilidade. Assim, a classe *Product* se relaciona com alguma classe *Compatibility*, que então se relaciona com a classe *ConsumerItem* (cf. Figura 3).

Neste caso, a representação de uma compatibilidade é efetuada usando uma instância da classe de tipo de compatibilidade. Um produto deverá se relacionar com uma instância do tipo de compatibilidade que deseja-se representar; e essa instância se relaciona com o item do consumidor ao qual a compatibilidade se refere. O código 3 apresenta exemplos detalhados.

Essa solução é capaz de representar todos os tipos de compatibilidades contempla-

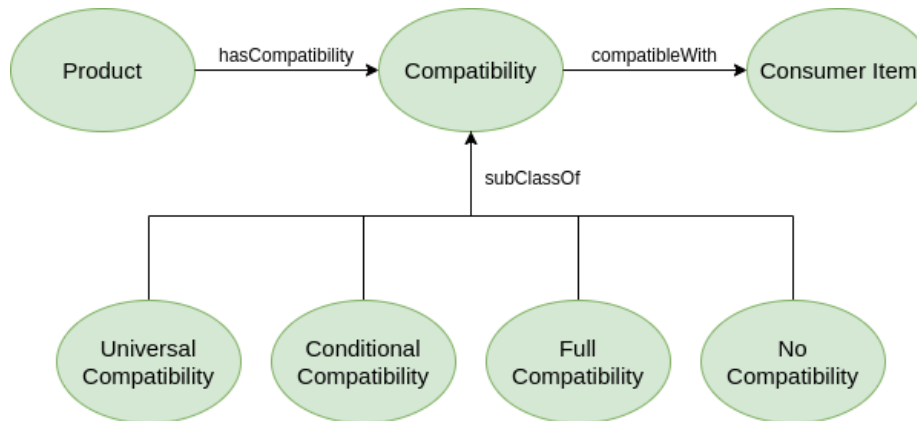


Figura 3: Representação do conceito de compatibilidade como uma classe.

dos pela primeira abordagem. É igualmente capaz de representar compatibilidades universais, que seriam representadas por uma nova subclasse de Compatibilidade. Dessa forma, quando houver um produto universal, é criada uma instância de uma compatibilidade universal e ela é relacionada ao produto, da mesma forma que é feito com outros tipos de compatibilidade. A única diferença é que agora não haverá necessidade de a instância da compatibilidade se relacionar com nenhum item do consumidor, pois não faz sentido quando se trata de uma compatibilidade universal. Um exemplo também pode ser visto no código 3.

Agora é usada uma instância para representar uma compatibilidade entre produto e item do consumidor. Isso possibilita que propriedades sejam adicionadas a essa instância, e por consequência, cada relação produto-carro pode ter informações próprias. Essa característica resolve um outro problema da primeira abordagem, permitindo adição de metadados sobre cada compatibilidade.

Para ilustrar as novas possibilidades descritas, assim como demonstrar uma implementação da nova solução, o código 3 representa, usando essa nova abordagem, as mesmas compatibilidades representadas no código 2. Além delas, representa também um *product3* que corresponde a um produto de compatibilidade universal e também a data em que essa compatibilidade foi computada.

Essa abordagem é mais verbosa e consideravelmente menos intuitiva em comparação com a primeira, embora seja mais completa por resolver os problemas levantados. Há uma grande diferença no número de triplas: cada relação de compatibilidade, que na primeira abordagem era representada apenas como uma tripla, nessa abordagem é, em geral, representada usando 3 triplas: uma instanciando a compatibilidade com o devido tipo, uma relacionando o produto à compatibilidade, e uma terceira relacionando a compatibilidade com o item do consumidor.

```

onto:hasCompatibility rdf:type owl:ObjectProperty .
onto:compatibleWith rdf:type owl:ObjectProperty .
onto:Compatibility rdf:type owl:Class .
onto:FullCompatibility rdf:type owl:Class ;
    rdfs:subClassOf onto:Compatibility .
onto:NoCompatibility rdf:type owl:Class .
    rdfs:subClassOf onto:Compatibility .
onto:ConditionalCompatibility rdf:type owl:Class .
    rdfs:subClassOf onto:Compatibility .
onto:UniversalCompatibility rdf:type owl:Class .
    rdfs:subClassOf onto:Compatibility .
onto:date_retrieved rdf:type owl:DataProperty .

graph:product1 rdf:type onto:Product .
graph:product2 rdf:type onto:Product .
graph:product3 rdf:type onto:Product .
graph:car1 rdf:type onto:Car .
graph:car2 rdf:type onto:Car .

graph:comp_prod1_car1 rdf:type onto:FullCompatibility .
graph:product1 onto:hasCompatibility graph:comp_prod1_car1 .
graph:comp_prod1_car1 onto:compatibleWith graph:car1 .
graph:comp_prod1_car2 rdf:type onto:ConditionalCompatibility .
graph:product1 onto:hasCompatibility graph:comp_prod1_car2 .
graph:comp_prod1_car2 onto:compatibleWith graph:car2 .
graph:comp_prod2_car1 rdf:type onto:FullCompatibility .
graph:product2 onto:hasCompatibility graph:comp_prod2_car1 .
graph:comp_prod2_car1 onto:compatibleWith graph:car1 .
graph:comp_prod2_car2 rdf:type onto:NoCompatibility .
graph:product2 onto:hasCompatibility graph:comp_prod2_car2 .
graph:comp_prod2_car2 onto:compatibleWith graph:car2 .
graph:comp_prod3 rdf:type onto:UniversalCompatibility .
graph:product3 onto:hasCompatibility graph:comp_prod3 .

graph:comp_prod3 onto:dateRetrieved "2020-08-16"^^xsd:date .

```

Código 3: Exemplo de triplas RDF para o caso de usarmos compatibilidades como classes. Elas representam as mesmas compatibilidades representadas no código 2, e ainda um novo *product3* que corresponde a um produto de compatibilidade universal que também conta com a data em que essa compatibilidade foi computada.

4 Discussão

Este trabalho propôs maneiras de estruturar conhecimento de compatibilidades no contexto de *e-commerce*, em especial relacionando produtos de lojas do setor automobilístico e carros, que podem ou não ser compatíveis com produtos à venda nas lojas online. Levamos em conta diferentes tipos de compatibilidades e também características práticas de cada representação considerando necessidades presentes na *GoBots*.

Desenvolvemos uma ontologia para representar compatibilidades no domínio automobilístico. No entanto, o conceito de Item do Consumidor garante flexibilidade à modelagem apresentada. A mesma ontologia poderia também ser utilizada para representar compatibilidades em outros domínios para além do automotivo, sendo necessário apenas a criação de uma nova subclasse de Item do Consumidor.

Abordamos com profundidade as representações de compatibilidade entre produtos e itens do consumidor. Essas representações também dependem indiretamente de formas de representar e identificar instâncias únicas desses dois conceitos. Como já mencionado, produtos em *e-commerces* geralmente possuem identificadores únicos já atrelados a eles, mas existem diferentes identificadores. Talvez o mais comum deles seja o EAN, que representa o código de barras e se compromete a representar universalmente um produto. Mas nem todas as lojas de *e-commerce* identificam seus produtos com EAN, muitas delas identificam com códigos internos, interpretáveis apenas pela loja e não necessariamente únicos, como o SKU. Para representar instâncias de produtos, portanto, é necessário não só conhecer esses diferentes identificadores como também administrá-los de modo a evitar instâncias repetidas de mesmos produtos representados com identificadores diferentes.

Da mesma forma, também é preciso identificar instâncias únicas de itens do consumidor. Já foi discutido anteriormente que a identificação desses itens dependem de atributos ditados pelo domínio dos itens. Por exemplo, itens do domínio automotivo atualmente são identificados na ontologia pelos atributos modelo, marca e ano. Esses são tidos como atributos essenciais para identificar um carro, mas também existem outros atributos, como número de portas, potência do motor ou tipo de câmbio, que podem ser úteis ou até necessários para determinar compatibilidades com determinados produtos ou categorias de produtos. É preciso então estudar técnicas para manipular essa diversidade de atributos de modo que seja possível diferenciar itens do consumidor que possam de fato apresentar compatibilidades diferentes, mas também evitar repetições desnecessárias de instâncias que vão ter sempre as mesmas compatibilidades.

As propostas de modelagem sugeridas foram desenvolvidas com base no padrão RDF da web semântica, que impõe que grafos de conhecimentos sejam baseados em triplas sujeito-predicado-objeto. No entanto, uma nova proposta tem sido avaliada pela W3C que poderia gerar novas possibilidades de representação de compatibilida-

des. Essas se mostram como potenciais próximas linhas de pesquisa neste trabalho. O RDF* [3] é uma extensão do RDF que tornaria possível representar grafos RDF complexos de maneira mais intuitiva. Em especial, ele tornaria possível a adição de metadados às propriedades tradicionais, o que é especialmente interessante para a ontologia desenvolvido nesta investigação. Essa impossibilidade no RDF é um dos principais problemas da primeira abordagem apresentada.

5 Conclusão

Representar e estruturar informações de compatibilidade se mostra uma tarefa relevante para o contexto de *e-commerce* e envolve diferentes problemáticas. Esse trabalho analisou relações de compatibilidade e investigou meios de representar compatibilidade de maneira efetiva de forma que possa ser facilmente recuperadas em grafos de conhecimento. Apresentamos e detalhamos os conceitos fundamentais que fazem parte de uma compatibilidade, assim como diferentes abordagens sobre como modelá-la em uma ontologia usando RDF. A modelagem final apresentada tem sido explorada no mundo real pela *GoBots* para estruturar compatibilidades de produtos no domínio automotivo. O modelo vem sendo validado à medida que auxilia a agrupar cada vez mais informações. Essas, por sua vez, são usadas para melhorar a efetividade da geração de respostas automáticas para perguntas sobre compatibilidade.

Referências

- [1] Biralatei Fawei, Jeff Z Pan, Martin Kollingbaum, and Adam Z Wyner. A semi-automated ontology construction for legal question answering. *New Generation Computing*, 37(4):453–478, 2019.
- [2] T. R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.
- [3] Olaf Hartig and Bryan Thompson. Foundations of an alternative approach to reification in rdf. *arXiv preprint arXiv:1406.3399*, 2014.
- [4] Markus Lanthaler, Richard Cyganiak, and David Wood. RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C, February 2014. <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>.
- [5] Taehee Lee, Ig-hoon Lee, Suekyung Lee, Sang-goo Lee, Dongkyu Kim, Jonghoon Chun, Hyunja Lee, and Junho Shim. Building an operational product ontology system. *Electronic Commerce Research and Applications*, 5(1):16–28, 2006.

- [6] Diogo Teles Sant’Anna, Rodrigo Oliveira Caus, Lucas dos Santos Ramos, Victor Hochgreb, and Julio Cesar dos Reis. Generating knowledge graphs from unstructured texts: Experiences in the e-commerce field for question answering. In *Advances in Semantics and Linked Data: Joint Workshop Proceedings from The International Semantic Web Conference - ISWC 2020*, pages 56–71, 2020.
- [7] Pedro Szekely, Craig A. Knoblock, Jason Slepicka, Andrew Philpot, Amandeep Singh, Chengye Yin, Dipsy Kapoor, Prem Natarajan, Daniel Marcu, Kevin Knight, David Stallard, Subessware S. Karunamoorthy, Rajagopal Bojanapalli, Steven Minton, Brian Amanatullah, Todd Hughes, Mike Tamayo, David Flynt, Rachel Artiss, Shih-Fu Chang, Tao Chen, Gerald Hiebel, and Lidia Ferreira. Building and using a knowledge graph to combat human trafficking. In *Proceedings, Part II, of the 14th International Semantic Web Conference on The Semantic Web - ISWC 2015 - Volume 9367*, pages 205–221, 2015.