Knowledge Graph-based Product Recommendations on e-Commerce Platforms

André Gomes Regino[1][a], Rodrigo Oliveira Caus[1], Victor Hochgreb[1][b] and Julio Cesar Dos Reis[2][c]
[1]*Institute of Computing, University of Campinas, Campinas, São Paulo, Brazil*
[2]*GoBots, Campinas, São Paulo, Brazil*

Keywords: Recommendation Systems, Knowledge Graphs, Question Answering Systems.

Abstract: The amount of data generated in e-commerce sales has expressively grown in the last few years. Online stores often receive questions about products related to price, guarantee, and shipping price. By reducing time for prompt answering, stores can improve customer satisfaction and sales conversion rate. The recommendation of available alternative products in case of product unavailability intended by the customer plays a key role in sales growth in this context. This article defines and evaluates a technique for product recommendation based on the product's facts stored in Knowledge Graphs (KGs). Our KG is filled with facts from natural language questions and answers processed from the e-commerce platform. We exemplify our proposal in a real-world solution, using data from online stores processed by GoBots, a leading e-commerce chatbot business in Latin America. Online sellers assessed the results of the recommendations to evaluate their quality.

1 INTRODUCTION

The convenience of online stores has captured customers' attention who, a few decades ago, only made face-to-face purchases in commercial establishments. The COVID-19 pandemic has changed customers' relationships with such online stores because they were one of the few ways to buy products. The virtual stores, called e-commerces, had to adapt to this growing demand. In this context, the guarantee of security and integrity in online sales should be allied to fair prices and deliveries in a reasonable time. These aspects guarantee a better customer experience throughout the buying process.

Product recommendation figures as a key strategy to offer a complete service to the final customer. This aims to deliver one or more products that suit the customer's taste or need. The suggested product should be compatible with the customer's purchasing behavior or characteristics. An adequate recommendation is an ally when the required product is not in stock or is no longer traded. In this case, a sale compatible with the customer's needs would please both the customer and the seller. In this scenario, the customer would not leave without the purchase, and the seller would not miss a sale.

The recommendation system functions as an information filtering. In the e-commerce scenario, the aim is to filter products that may be interesting to a given customer (Shao et al., 2021), choosing products when there are many available options (Isinkaye et al., 2015). Amazon, a big player in the e-commerce scenario, increased its sales by 35% after adopting a recommendation system to display specific products to its customers (Lee and Hosanagar, 2014). However, recommending a product is not an easy task. It depends on the recommendation strategy adopted to be successful.

Existing approaches use different data sources to identify the compatibility between the customer purchase history and the catalog of items in the e-commerce platforms. Examples of these data are product ratings, attributes, and user search history (Dwivedi et al., 2020). The strategy can vary from collaborative filtering (Linden et al., 2003), machine learning (Covington et al., 2016) and Knowledge graph-based techniques (Guo et al., 2020). Lately, knowledge graphs (KGs) have been studied and explored for recommendation purposes. For example, AliCoco, a KG used in the largest e-commerce in China, AliBaba (Luo et al., 2020). KGs are helpful to represent connections between customers, products,

[a] https://orcid.org/0000-0001-9814-1482
[b] https://orcid.org/0000-0002-0529-7312
[c] https://orcid.org/0000-0002-9545-2098

Regino, A., Caus, R., Hochgreb, V. and Reis, J.
Knowledge Graph-based Product Recommendations on e-Commerce Platforms.
DOI: 10.5220/0011388300003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 2: KEOD*, pages 32-42
ISBN: 978-989-758-614-9; ISSN: 2184-3228

and sales.

This article proposes a novel methodology relying on KGs to recommend products based on the compatibility between an item owned by the customer and the product being sold. Our solution differs from the state-of-art studies. Our approach performs product recommendations searching for compatibility connections represented (encoded) in a KG. The recommendation is based on a compatibility question made by the customer and the compatibility among the desired product (by the customer) and similar products indexed by our KG triples. No user data is required in our proposed recommendation process. Our KG was initially intended to answer compatibility questions made by customers in large Brazilian e-commerce platforms (Sant'Anna et al., 2020).

The contribution of this research is to further extend the system functionality by automatically providing answers with recommended products in case of incompatibility between the item owned by the customer and the desired product. To the best of our knowledge, there is no evidence in literature of a solution that recommends products in an e-commerce context based on a question-answering approach combined with the products' attributes representation. Our evaluation results based on sellers' opinions reveals that the suggested recommendations from the system are adequate. We experienced an increase in the number of recommendations provided by the system.

The remaining of this article is organized as follows: Section 2 discusses related work; Section 3 presents our proposed solution by including the formalization of our recommendation technique and a running example from a real-world e-commerce platform. Section 4 conducts an evaluation to assess sellers' perspectives and perceptions on the recommendation features and results in the platform. In section 5 we further discuss our findings. Section 6 presents the final remarks.

2 RELATED WORK

This section presents relevant approaches from state-of-the-art studies of knowledge-graph-based recommendation systems.

Guo *et al.* (Guo et al., 2020) gathered a set of 39 studies focused on KG-based recommendation systems. They classified the recommendation algorithms into three groups: the collaborative-filtering-based, the content-based, and the hybrid solutions. The collaborative-filtering aims to match the user profile with other profiles in the platform (e-commerce, video streaming, *etc*), matching their behavior. It is the most common recommendation method. In order to improve its efficiency and accuracy, several studies explored external sources to gather user data, such as social media and user location. The content-based approach uses past users' information and behavior tracking in the platform. The rationale is that users may be interested in products similar to those they already bought. The hybrid category addresses challenges found in both collaborative-filtering and content-based categories: the cold start, data sparsity, and heterogeneity, to cite a few. The authors also discussed a list of the main challenges in the relationship between KGs and recommendation systems. The time-consuming is a relevant aspect to take into account. These methods use machine learning algorithms, such as Convolutional Neural Networks (CNN), to train the recommendation model. In large online systems, the recommendations should be updated constantly, requiring a fully retraining of the deployed models. The authors indicated this task of model training and its overtime update as one of the significant challenges in the area.

AliCoCo (Luo et al., 2020) is a state-of-art KG that conceptualizes users' needs in Alibaba's platform. Their motivation relied on users' dissatisfaction with Alibaba's recommendation results, which generated much redundant display of products. To overcome it, a KG was built to encode brands, categories of products, and search concepts. Their study (Luo et al., 2020) found that customers in Alibaba's platform are interested in concepts like "Outdoor Family Barbecue". This consists of products of different types and brands (grill, knife, butter, sunblock), not only individual products. The reported results indicated an increase in the click-through rate and the customer satisfaction based on GMV (Gross Merchandise Value) rate.

Wu *et al.* (Wu et al., 2019) presented a summary of methods related to how natural language questions are answered based on KGs. The authors organized existing KGs by their domain application like open domain KGs, functioning as an encyclopedia for several domains (*e.g.*, DBpedia) and domain-specific KGs. We classify our solution as a niched category at the current stage, modeling the automobile domain. This specific domain was considered in our study due to the high amount of existing compatibility questions in the e-commerce platform. The RDF triples in our KG were constructed based on the processing of these questions (Sant'Anna et al., 2020).

In addition, according to Wu *et al.* (Wu et al., 2019), the second category organization relates to KGs parsing method by classifying them in two groups: semantic parsing and information retrieval. In semantic parsing, the questions are transformed into logical symbols to represent the semantics intended by the authors' questions. The disadvantage of semantic parsing is that it is highly dependent on external resources, such as dictionaries and labeling techniques, to categorize data modeled by the KG (Wu et al., 2019). On the other hand, the information retrieval extracts relevant parts of the questions to query the KG looking for candidate answers. After ranking the found answers, it creates an optimized final answer. Wu *et al.* (Wu et al., 2019) described that this technique has several performance issues when compared to semantic parsing methods.

Our proposed recommendation system is suited to automatically recommend products (represented in our RDF KG) in an e-commerce platform. We implemented it in a real-world software application running for online stores. Our main contribution refers to an original recommendation technique fully developed that explores compatibility facts encoded in our KG to produce recommendations based on products' attributes. We deployed and applied our system to an e-commerce platform in Brazil. To the best of our knowledge, there is no evidence in the literature of a solution that recommends products in an e-commerce context based on a question-answering approach combined with the products' attributes representation.

3 RECOMMENDATION SYSTEM FOR E-COMMERCE

This section describes an overview of our recommendation system (cf. Subsection 3.1). Subsection 3.2 further formalizes and presents details of our specific KG-based recommendation used to suggest products in e-commerce platforms. Subsection 3.3 shows a running example of a recommendation proposed by our system.

3.1 Overview

A KG is a directed graph with nodes representing real-world entities such as "Rio de Janeiro"; and edges representing relations between entities; *e.g.*, "Rio de Janeiro" "countryName" "Brazil". KGs have the form of RDF triples in a way that $G = t_1, t_2, ..., t_n$. A RDF triple (t) refers to a data entity composed of subject (s), predicate (p) and object (o) defined as $t = (s, p, o)$.

Sant'Anna *et al.*(Sant'Anna et al., 2020) introduced the KG used in this investigation. Our previous work addressed how to generate triples by processing pairs of customer's questions and attendant's answers in natural language. The encoded knowledge was used to automatically answer customer's compatibility questions, indicating whether the product fits in their vehicles. This KG answered more than 31,000 compatibility questions of 77 stores in one of the biggest Latin American marketplaces.

Our KG expresses knowledge on compatibility between products of the automotive domain of e-commerce and cars. Figure 1 shows an example of a set of triples in our KG. It presents an instance of the class Store (*car-parts*) that sells an instance of the class Product (ID "108093"). A consumer item of the class "Car" ("fiesta-2012") is fully compatible with the instance of the Product class.

Figure 2 presents the key steps in the recommendation process designed in our approach based on a compatibility question performed by a customer (formalized in Algorithm 1). The input is a compatibility natural language question that a customer asks on a product page. The output is an answer containing a list of recommended products.

First, the e-commerce user (customer) asks a question q written in a natural language text about a product p. The question q is processed by a GoBots service that implements an intent and entity discovery process based on question q as input (step A in Figure 2).

At this step (Line 1 in Algorithm 1), the entities $E = \{e_1, e_2, ..., e_n\}$ from the question are identified (*e.g.*, "brand" and "model" are used to identify attributes of a car). Algorithm 1 identifies the intention i of the question (Line 2). Our recommendation algorithm is suitable for questions of compatibility intention (Line 3 in Algorithm 1). A customer asking whether the product in question, which may be a car part, fits in a particular vehicle is a question with compatibility intent.

If the intention i is related to "compatibility" (Line 3 of Algorithm 1), our algorithm proceeds to the second step (Step B in Figure 2). It evaluates whether the item in the consumer's possession ci (called "consumer item" from now on) is compatible with the product sold p (Line 5 of Algorithm 1).

The function "getCompatibility" queries the KG retrieving a response r_c. The response $r_c = \{f, t, a_{com}\}$ from the KG is composed by a boolean f, indicating whether the compatibility was found or not; a compatibility type t and an attendant answer a_{com}. There are four types of valid compatibility, all of them representing disjoint subclasses of the Compatibility KG class:

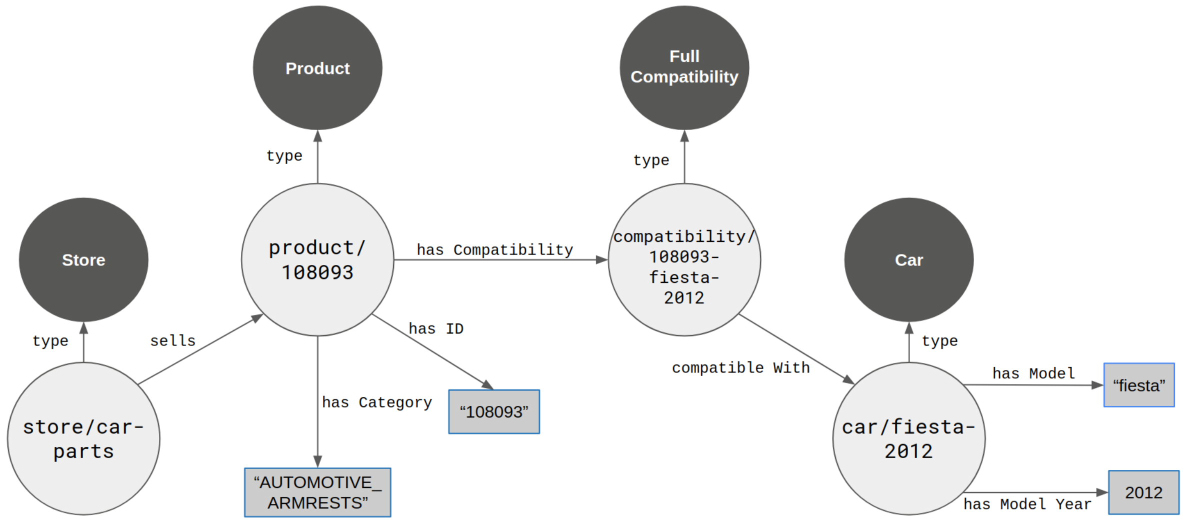

Figure 1: Example of a set of triples registered in our KG for complete compatibility facts between a product and the car "Fiesta 2012". The darker nodes represent classes and the lighter nodes represent instances.

$$T = \{FullCompatibility, UniversalCompatibility, ConditionalCompatibility, NoCompatibility\} \quad (1)$$

The types of compatibility indicate that the product fits with the consumer item (Full Compatibility); partially fits (Conditional Compatibility); does not fit (No Compatibility); fits with any consumer item (Universal Compatibility). An example of a Full Compatibility type can be a given car exchange perfectly fitting in a "Honda Civic 2012".

Algorithm 1: Product recommendation based on a consumer item and a desired product.

Require: $q, recommendedProducts, p$
1: $E \leftarrow identifyEntities(q)$
2: $i \leftarrow identifyIntent(q)$
3: **if** $i \equiv$ "$compatibility$" **then**
4: $ci \leftarrow extractConsumerItem(E)$
5: $r_c \leftarrow getCompatibility(ci, p)$
6: **if** $r_c.t \equiv NoCompatibility$ **then**
7: $recommendedProducts \leftarrow Algorithm2(ci, p)$
8: **return** $recommendedProducts$
9: **else**
10: **return** $r_c.a_{com}$
11: **end if**
12: **end if**

If $t = NoCompatibility$ (Line 6), indicating that ci is not compatible with p, Algorithm 1 proceeds to the third step (output False of Step B in Figure 2 leading to Step C). At this point, a new request is performed to the KG to identify which products from the seller store $P_{rec} = \{p_{rec1}, p_{rec2}, ..., p_{recn}\}$ fit the consumer item ci (*e.g.*, an older version of the car exchange may fit in the Honda Civic 2012). Section 3.2 presents this procedure in more details. In case that there are no available compatible products to be recommended, the procedure returns an answer stating that ci ad p are not compatible (output False of step C in Figure 2).

In case that $t \neq NoCompatibility$ ($t = FullCompatibility$ or $t = ConditionalCompatibility$ or $t = UniversalCompatibility$), the algorithm identifies that the ci and p are compatible and returns the answer that states the compatibility between the items (Lines 9 and 10 of Algorithm 1).

3.2 Recommendation Technique

This subsection provides a deeper description of Step C as the core of our recommendation system. In our approach, the rationale is that ci is not compatible with p, but there might be a set of products P_{rec} presented in the KG that fits ci. The difference between the KG requests from steps B and C is that in B, our algorithm verifies the compatibility between the pair (p, ci), whereas in step C, the request queries which products $p_{rec} \in P_{rec}$ fits ci. In addition, the customers can be interested in other products of the same seller, but they do not know which seller's products fit in their consumer item.

The set of recommended products P_{rec} is composed based on inclusion criteria I. The motivation is the necessity to retrieve a well-defined set of available products, meeting a pre-defined standard. The following items show the criteria adopted by our algorithm to recommend products:

1. $\forall\ p_{rec} \in P_{rec}$ and p should have the same category;
2. $\forall\ p_{rec} \in P_{rec}$ should be compatible with consumer item ci;
3. $\forall\ p_{rec} \in P_{rec}$ and p should be offered in the same store;
4. $\forall\ p_{rec} \in P_{rec}$ should be available by any customer to buy;
5. There must be one or more $p_{rec} \in P_{rec}$;

The first and second criteria show that the recommended products should have the same category as the product asked by the customer. They should fit in the consumer item described in the user's question (*e.g.*, only car exchanges should be recommended to the user if the compatibility question relates to a car exchange fitting a "Honda Civic 2012" car). The third criterion is that only products sold by the seller should be recommended. In an e-commerce platform, many sellers are competing for a market share, and recommending the competitor's products would be unacceptable. The fourth and fifth criteria are the status of the product campaign. In other words, if the product is being sold and more than one unit of it is available. Recommending a product out of stock could negatively affect the seller's reputation. The data required in some items of I (category, store, product availability) is retrieved using the seller's marketplace APIs.

At Step C, Algorithm 2 requires ci, p and I, and outputs an object of the class Recommendation (r), composed by a boolean value indicating that the recommended products were found ($r.found$) and a recommendation answer (*r.answer*). Algorithm 2 uses a SPARQL query (Line 2) to get a preliminary list of products P_{rec} registered in KG that have a *FullCompatibility* relation with ci. The Query 1 presents a SPARQL template for the automotive domain. The literal "PRODUCT_CATEGORY" represents the title of the category attribute of p; the STORE_NAME resource represents the name of the store that sells p; "CAR_MODEL" and "CAR_YEAR" literals represent the attributes model and year of ci, respectively. Each resource or literal on this template is replaced by its respective value obtained from p and ci.

```
BASE <http://kg.test/KB/knowledgegraph
    />
PREFIX onto: <http://kg.test/KB/
    ontology>
SELECT ?id ?compatibility FROM <http://
    kg.test/KB>
WHERE {
    ?product rdf:type onto:Product .
    ?product onto:hasCategory "
        PRODUCT_CATEGORY" .
    ?product onto:hasID ?id .
    ?product onto:hasCompatibility ?
        compatibility .
    <STORE_NAME> onto:sells ?product .
    {
        ?compatibility onto:
            compatibleWith ?car .
        ?car onto:hasModel "CAR_MODEL"
            .
        ?car onto:hasModelYear "
            CAR_YEAR" .
        ?compatibility rdf:type onto:
            FullCompatibility
    }
}
```

Query 1: SPARQL template to retrieve P_{rec} based on ci and p category.

The list of recommended products retrieved by Query 1 is filtered based on the remaining items of inclusion criteria I (Line 3 in Algorithm 2). After querying and filtering the set of products P_{rec}, Algorithm 2 applies a ranking strategy to delimit which products might be more interesting for recommendation to the consumer. We explore the conversion rate ($conv$). This rate is composed of the number of sales (s) divided by the product's visits (v). P_{rec} is ordered by conversion rate in descending order (*i.e.*, the higher the conversion, the higher the product is in the ranking). The conversion rate is described in Equation 2. At this stage, we could also rank products by sales, but this would penalize recently added products in the e-commerce platform.

$$conv(p) = \begin{cases} s(p)/v(p) & \text{if } v > 0 \\ \mathbf{0} & \text{otherwise} \end{cases} \quad \forall p \in P_{rec} \tag{2}$$

At this point, there is a set of filtered and ranked recommended products (Lines 3 and 6 in Algorithm 2). Additionally, we opted to limit the number of results. The top k products form the final recommendation list stored in P_{rec}. In case of recommended products ($P_{rec} \neq \emptyset$), our solution returns a range of recommended products (Line 7 of Algorithm 2). The last step is to build an answer *r.answer* containing the recommended products P_{rec} links (Line 11 in Algorithm 2).

3.3 Running Example

This section presents a real-world running example extracted from our KG (cf. Figure 3). Our KG is constantly updated, reaching a top rate of 2,000 newly added nodes per day. It has approximately 500,000 compatibility nodes and more than 3,500,000 triples. This KG is used to answer real-time compatibility

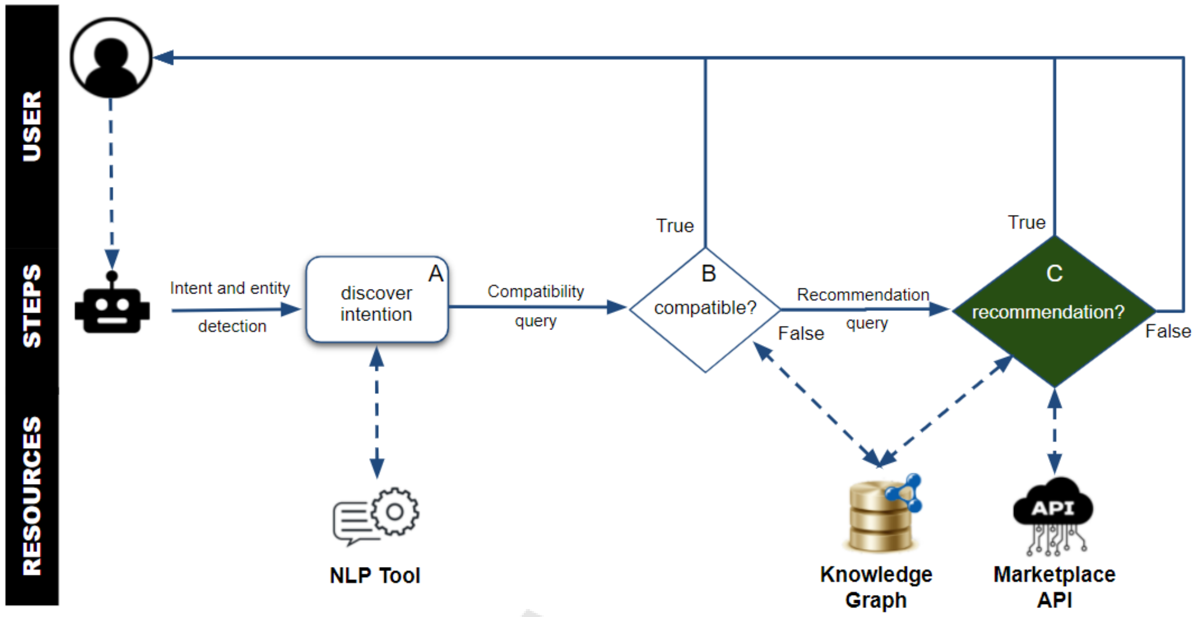

Figure 2: Recommendation based on question-answer facts.

Algorithm 2: Recommendation procedure.

function $recommendation(ci, p)$
  Require: ci, p, I, k
  $P_{rec} \leftarrow constructSparqlQuery(ci, p)$
  $P_{rec} \leftarrow filterProducts(ci, I)$
  if $P_{rec} \neq \emptyset$ **then**
    $r.found \leftarrow True$
    $P_{rec} \leftarrow rankProducts(P_{rec})$
    $P_{rec} \leftarrow limitProducts(P_{rec}, k)$
  else
    $r.found \leftarrow False$
  end if
  $r.answer \leftarrow buildAnswer(P_{rec})$
  return r

questions regarding automotive domain from customers in a big Latin-American marketplace. In the scenario described in Figure 3, a product p (speaker) was found in an online automobile store in one of the largest Latin America marketplaces. Initially, a customer enters the speaker product page and asks a question q. The customer is in doubt if the speaker can be used in his car, a “Corsa 2004” model (ci). In the Portuguese language, the customer creates a question: “Can I use this speaker in my Corsa 2004 car?” (Question translated to English by the authors).

This scenario shows an example of a question that can result in a recommendation answer. The intent (i) of the question is compatibility. “Corsa” is the car model, and “2004” refers to the manufacturing year of the car. Both characteristics form the definition of ci. In this case, a car; and p is represented by a speaker in stock. The question arrives at the GoBots service in a textual format. The service is responsible for identifying both i and E of q (Step A in Figure 2). The NLP tool identifies that the question has a compatibility intent and some entities, including a model and a year of a car product.

This data is sent to the KG service (Step B in Figure 2) responsible for handling events related to the KG. A SPARQL query is formed to verify if there is a compatibility between ci (“Corsa 2004”) and p (“speaker”). This SPARQL query returns from the KG whether the items are compatible. In our case, the KG responded with compatibility type (t) equals to “*NoCompatibility*”. Based on this response, our algorithm proceeds to the recommendation step (C in Figure 2). A new SPARQL query is formed, using as parameter ci (“Corsa 2004”) and p (“speaker”).

Query 2 presents the recommendation query submitted to our KG related to our running example. It selected products sold by a store (<car-sound-parts>) presented in the same category of the speaker (“AUTOMOTIVE_SPEAKER”) that are compatible with a specific product (“car”), model (“Corsa”) and year (“2004”).

```
BASE <http://kg.test/KB/knowledgegraph
    />
PREFIX onto: <http://kg.test/KB/
    ontology>
SELECT ?id ?compatibility FROM <http://
    kg.test/KB>
WHERE {
    ?product rdf:type onto:Product .
    ?product onto:hasCategory "
```

```
        AUTOMOTIVE_SPEAKER" .
    ?product onto:hasID ?id .
    ?product onto:hasCompatibility ?
        compatibility .
    <store/car-sound-parts> onto:sells
        ?product .
    {
        ?compatibility onto:
            compatibleWith ?car .
        ?car onto:hasModel "corsa" .
        ?car onto:hasModelYear 2004 .
        ?compatibility rdf:type onto:
            FullCompatibility
    }
}
```

Query 2: SPARQL query responsible for selecting automotive speakers sold by the store compatible with Corsa 2004

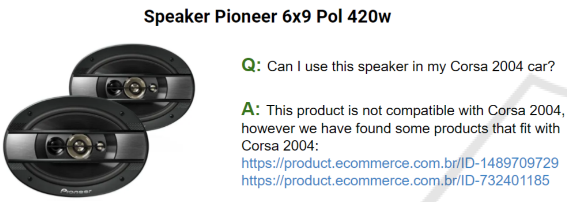

Figure 3: Running example of product recommendation. The original question was written in Portuguese and translated to English. The answer contains links to recommended products.

If no recommendation is found from our KG triples, the designed Algorithm returns a text message to the user informing that "This product is not compatible with Corsa 2004". In our example, the KG returned a not empty P_{rec} set. This set is filtered using I and ranked by conversion (sales/visits). Speakers out of stock are removed from P_{rec}. In this case, our Algorithm returns the text: "This product is not compatible with Corsa 2004. However, we found some products that fit Corsa 2004", followed by a list of links. In our running example, there are two recommended products.

4 EVALUATION

This section describes our evaluation about the recommendation feature. This feature was activated for 50 customer stores of GoBots; all of them related to the automobile domain. Subsection 4.1 describes a quantitative analysis based on system logs whereas Subsection 4.2 presents results of the qualitative analysis relying on collected data from sellers' opinions.

4.1 Quantitative Evaluation

We conducted the quantitative analysis by logging the number of recommendations produced by our solution. Among the stores that had the functionality activated, we logged how many responded with recommendations, and how many had positive results from the sellers' perspective.

We gradually activated the system for online stores in the automotive sector from November 2021 to April 2022. At the end of this period, the system had automatically responded with product recommendations on thirty one (31) stores. Figure 4 shows three graphics related to the quantitative analysis. Stores that have not yet responded to compatibility questions may be small stores with few products and few questions.

Of the 31 stores, there were a total of 457 responses with recommendations containing links to other products (graphic 1 in Figure 4); 171 other responses indicated no compatibility between the customer's product and the store's product; and no available product to be recommended. This resulted in a total of 608 responses related to the recommendation (graphic 2 in Figure 4).

The total value presented in graphic 1 in Figure 4 (457) corresponds to all responses with at least one recommendation; those that followed the "True" path in the decision flowchart of step C in Figure 2. At least 457 items were suggested to users who may have purchased the recommended item. On the other hand, the total value presented in graphic 2 in Figure 4 (628) corresponds to all responses that included or not a recommendation; those that followed the path "True" or "False" in the decision flowchart of step C of Figure 2.

Almost half of the recommendations from both graphic 1 and graphic 2 (in Figure 4) are from a single store, which is one of the biggest and with highest sales in the e-commerce platform where this study is situated. Graphic 3 (in Figure 4) reveals an increase in the number of responses over the weeks analyzed. This result was due to both the vertical scalability of the results (number of products and recommendations growing) and the horizontal scalability (number of activated online stores using the recommendation feature).

The proportion between the total expressed in Graphic 1 and in Graphic 2 in Figure 4 shows an interesting metric. This is the number of recommendation responses divided by the maximum number of possible recommendation responses. The closer to 1, the more this number demonstrates that the sellers had products in stock compatible with the consumer's

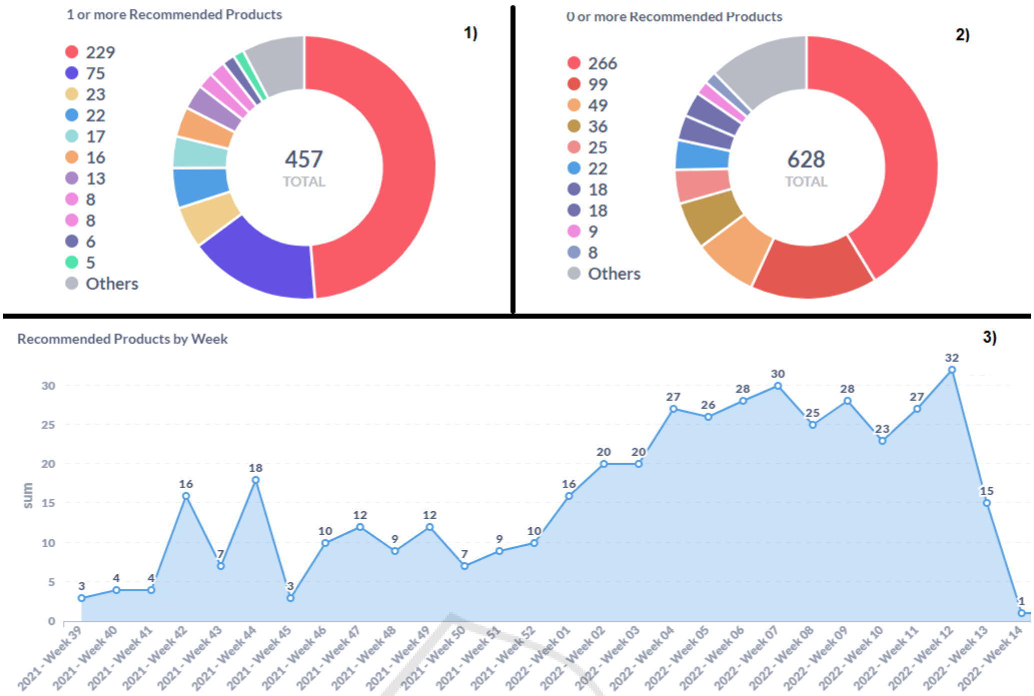

Figure 4: Sum of recommended products in three different perspectives. Graphic 1 shows that 457 responses contain at least one recommended products; Graphic 2 presents 628 responses containing 457 responses from Graphic 1 plus 171 responses with no products available to be recommended; Graphic 3 shows the increase of the number of recommendation answers by several weeks of the year.

item. A value close to 0 indicates that, although the recommendation functionality is active, no products compatible with the consumer's item are found.

The sellers have another way of giving feedback on some customer responses. If the seller finds any kind of error, whether it's in the recommended links or in the answer as a whole, he can mark it as incorrect. We identified that among the 457 recommendations, only 7 (5.87%) were marked by the sellers as incorrect. This shows the quality of the recommendation system, correctly answering 94.13% of the questions with recommendation answers performed.

4.2 Qualitative Evaluation

Our qualitative evaluation was conducted in two steps. We first proposed and sent an online form to a seller to obtain initial feedback regarding the recommendation. This seller was chosen because it represents the store with more recommendation responses and more errors pointed out by the seller. Afterwards, we conducted an online interview with this seller to further comprehend the perception regarding the recommendation feature in the e-commerce platform. We aimed to figure out the seller's assessment regarding the recommendation results.

We presented and analyzed two real-world recommendation answers for different products generated by our solution regarding the recommendation feature based on the KG. The first product is a "Rear View Camera for the Ecosport car 2020 model" (translated to English language by the authors). A user asked the following question: "Does it fit in Ford Ka 2019?" (translated to English language by the authors). Our algorithm 1 identified the compatibility intent and "Ford Ka 2019" as a car.

The second product was a "Screen Unlock Interface without DVD player for the Audi S4 car" (translated to English language by the authors). Another different user asked the following question: "Hi, I have an Audi S4 2013 without DVD Player, does it fit?" (translated to English language by the authors). Our algorithm 1 identified the compatibility intent and "Audi S4 2013" as a car. The seller answers the questions based on these two recommendations.

Figure 5 presents the designed feedback form in an infographic format. Our purpose was to evaluate both the idea of recommending products from the seller's point of view and the results obtained, including strengths and aspects that should be improved. The form was composed of 10 questions, 9 containing multiple-alternative answers; 2 of these questions were related to the general perception of the recommendation feature (section General of Figure 5). The others eight questions were related to the analysis of the results of recommendations (red and blue columns of Figure 5). The rationale behind our evaluation was to understand the recommendation's qualitative aspects in an easy-to-follow process to the seller.

As a result, the seller considered the recommendation feature useful (question 1 of Figure 5). In the second question the user had the opportunity of suggesting what improvements could be accomplished in the recommendation feature.

At this point, the seller provided his feedback about the first (questions 3 to 6) and second (questions 7 to 10) recommendations.

The seller answered that all products were correctly recommended (question 3) in a satisfactory way (question 4), respecting a good order of the links (question 5). The seller indicated the number of recommended items (3) unsatisfactory, since two links were equal (question 6).

For the second product recommendation, the seller informed that none of the products were correctly recommended (question 7), leading to unsatisfactory answers regarding the recommendation's order (questions 8 to 10). In this scenario, the seller informed that none of the products were specific to the consumer item (Audi A4 2013).

After reaching these results, we conducted a technical exploratory analysis to identify why the first recommendation was successful and the second one unsuccessful. In the first recommendation, the user asked about compatibility with the car, a "Ford Ka 2019". The recommended products were all fully compatible ($t = FullCompatibility$) with the car as input from the question. The recommendation feature led the user from an incompatible product ("Rear View Camera for the Ecosport car 2020 model") to products that fit the car. This could have generated a sale. Also, this fitted in the consumer item and was sold by the same store as the "Rear View Camera". If the recommendation feature were not activated at the time of the question, the answer would be that the product is not compatible, making the sale unfeasible.

The second recommendation was considered unsuccessful based on the seller's feedback. The user (customer) asked for compatibility with his car, an "Audi S4 2013". They received three products that had in their titles cars different from the consumer item (*e.g.,* "Screen Unlock Interface for Corolla Cross 2022"). These three products were of the "Universal Compatibility" type ($t = UniversalCompatibility$), which states that they (three different types of Screen Unlock Interface) fit in any car.

We found that the difference between the first and the second recommendation was the type of compatibility. Customers and sellers might not find interesting the recommendation of Universal Compatibility products.

Based on the exploratory analysis from the seller's answers, we found it necessary to go deeper into some aspects. We aimed to ensure that our perception regarding errors and successes in the recommendations was correct. We addressed it via an interview.

In the interview session (with the same seller that answered the feedback form), we first explained our purpose with the interview and the whole process of recommending one or more products to the seller. We explained that the recommendation feature in the e-commerce platform is only triggered if the product in stock is not compatible with the consumer item from the customer's query. We indicated that our goal was to determine whether the recommendation results were correct and to which extent they could be improved.

The interview was conducted in a semi-structured way based on three questions that summarized our doubts regarding the form answers given by the seller. The questions were related to universal compatibility; the number of recommended items; and ranking strategy. These topics were not explored in depth in the form step, and a direct conversation in the form of an interview with the seller would help solving open doubts.

Regarding the interview results, the first relevant point was the answer given to the question: "For your store, is it interesting that products that have universal compatibility are recommended?". The seller answered that "universal products" are great for the recommendation strategy, but his store had not many products of this type. Therefore, in this seller's context, there would not be a significant difference between recommending these types of products or not. In addition, the seller suggested that the appearance of these products should be highlighted to the consumer (on the webpage) if they were in the recommendation response.

Afterwards, we aimed to further understand the seller's affirmative (satisfactory) answer regarding the number of recommended items. The seller mentioned that he judged the number three as plausible because a higher number of recommendations would be too many. In this case, the answer would be polluted and difficult for customers to find the ideal product.

At the end of the interview, we approached the seller to obtain suggestions to improve the ranking strategy. The seller reaffirmed that the strategy used was sufficient and that other ranking strategies, such as ranking the products by their price, could generate unexpected effects. The seller cited, as an exam-

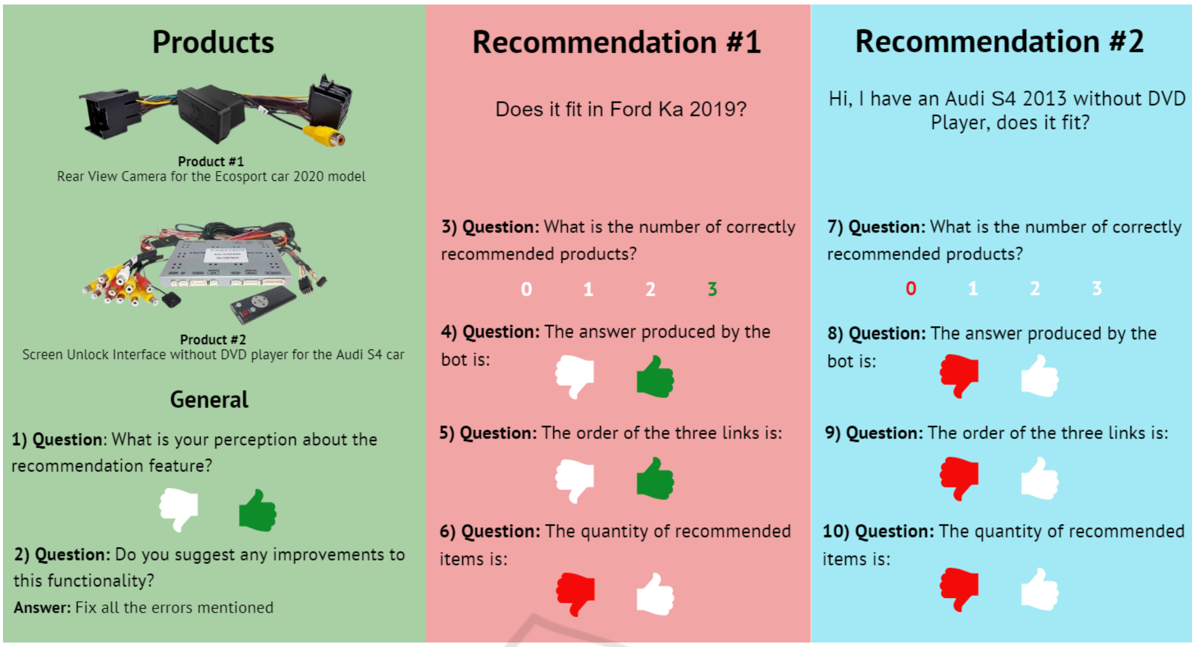

Figure 5: Infographic containing questions about the recommendation feature and answers of a Brazilian automobile seller where the recommendation feature is activated. The green column describes the 2 products used in the interview process and two questions related to both recommendations. The red and blue columns are related to the questions and answers about the first and second recommendation, respectively.

ple, a ranking based on product price, which could be an inefficient strategy because customers in his online store do not have preferences for more expensive and specific products or cheaper and generic products.

5 DISCUSSION

The recommendation of products in e-commerce is a complex task, given the diversity of users, their preferences, and the applicable domains in e-commerce platforms. Its importance is undeniable because products that would not be shown to the end users could be forgotten and not sold. We understand that the methodology presented in this research contributes to the state-of-the-art by appearing as a novel way of obtaining these types of recommendations. Relying on KGs and encoded compatibilities as RDF triples, we developed a methodology deployed in a real-world scenario.

Our qualitative evaluation conducted showed room for discussions and refinements around universal products and other decisions, such as the number of recommended items. These elements may vary according to the sellers' needs, and we understand that the seller can customize such decisions. The impact of these customizations as well as their implications are subject for future studies.

Our quantitative analysis reveled the effectiveness of our solution for automotive online stores in a great Latin American e-commerce platform. Via our solution and initial assessments, thirty-one online stores started to recommend products. This makes possible new sales that would have been ignored before, given the incompatibility between consumer's item and the product on sale. During the four months of our software tool monitoring, we observed an increase in recommendations per week. The value of 5.87% of incorrect recommendations, out of 457 recommendations made, indicates an acceptable error value by sellers.

We understand that our developed methodology and implemented techniques are easily expandable to other domains, as long as there is compatibility intention in texts used as input to the solution. Possible expansions of our investigation can be made by adding knowledge about compatibility between, for example, cell phones and their parts, clothes and people, among other domains.

An open challenge in our study remains to the use of data sources other than Q&A. Product descriptions, images, videos, external resources, and compatibility tables are examples of possibilities that can enrich KGs with further knowledge. As a consequence, our solution will be suited to generate more responses containing recommendations.

6 CONCLUSION

Having alternatives regarding products in e-commerce plays a key role for the purchase. However, finding and filtering adequate and compatible products based on customer's natural language questions remains a very difficult task. In this article, we proposed a novel methodology to recommend e-commerce products based on RDF KGs. The recommendation uses questions asked by e-commerce customers interested if their consumer item is compatible with the sales product. Our recommendation technique queries a KG filled with compatibility between consumer items and products. Our proposal was implemented and deployed in a real-world e-commerce platform in Latin America. We found that our solution is suitable for recommending relevant products in the e-commerce platform. Future work involves the study of novel methods to compare available products on e-commerce platforms. This work used textual properties, such as categories, store, and domain. The recommendation algorithm could further benefit from comparing NL texts and images, for instance. In addition, based on the evaluation results, we plan to further assess the use of products that are universally compatible within the recommendation feature.

ACKNOWLEDGEMENTS

This study was financed by the National Council for Scientific and Technological Development - Brazil (CNPq) process number 140213/2021-0.

REFERENCES

Covington, P., Adams, J., and Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 191–198, New York, NY, USA. Association for Computing Machinery.

Dwivedi, R., Anand, A., Johri, P., Banerji, A., and Gaur, N. (2020). Product based recommendation system on amazon data. *International Journal of Creative Research Thoughts - IJCRT*.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. (2020). A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge & Data Engineering*, (01):1–1.

Isinkaye, F. O., Folajimi, Y., and Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, 16(3):261–273.

Lee, D. and Hosanagar, K. (2014). Impact of recommender systems on sales volume and diversity. In Myers, M. D. and Straub, D. W., editors, *Proceedings of the International Conference on Information Systems - Building a Better World through Information Systems, ICIS 2014, Auckland, New Zealand, December 14-17, 2014*. Association for Information Systems.

Linden, G., Smith, B., and York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80.

Luo, X., Liu, L., Yang, Y., Bo, L., Cao, Y., Wu, J., Li, Q., Yang, K., and Zhu, K. Q. (2020). Alicoco: Alibaba e-commerce cognitive concept net. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 313–327.

Sant'Anna, D. T., Caus, R. O., dos Santos Ramos, L., Hochgreb, V., and dos Reis, J. C. (2020). 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 ISWC 2020*, pages 56–71.

Shao, B., Li, X., and Bian, G. (2021). A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Systems with Applications*, 165:113764.

Wu, P., Zhang, X., and Feng, Z. (2019). A survey of question answering over knowledge base. In Zhu, X., Qin, B., Zhu, X., Liu, M., and Qian, L., editors, *Knowledge Graph and Semantic Computing: Knowledge Computing and Language Understanding*, pages 86–97, Singapore. Springer Singapore.