QART: A Framework to Transform Natural Language Questions and Answers into RDF Triples

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: Semantic Web, Natural Language Processing, Text-to-Triples, E-Commerce.

Abstract: Knowledge Graphs (KGs) model real-world things and their interactions. Several software systems have recently adopted the use of KGs to improve their data handling. E-commerce platforms are examples of software exploring the power of KGs in diversified tasks, such as advertisement and product recommendation. In this context, generating trustful, meaningful and scalable RDF triples for populating KGs remains an arduous and error-prone task. The automatic insertion of new knowledge in e-commerce KGs is highly dependent on data quality, which is often not available. In this article, we propose a framework for generating RDF triple knowledge from natural language texts. The QART framework is suited to extract knowledge from Q&A regarding e-commerce products and generate triples associated with it. QART produces KG triples reliable to answer similar questions in an e-commerce context. We evaluate one of the key steps in QART to generate summary sentences and identify product Q&A intents and entities using templates. Our research results reveal the major challenges faced in building and deploying our framework. Our contribution paves the way for the development of automatic mechanisms for text-to-triple transformation in e-commerce systems.

1 INTRODUCTION

Recently, there has been digitalization of services previously performed exclusively in a physical way, such as retail commerce. E-commerces have become protagonists in sales. This change resulted in challenges to manage the massive amount of data related to the count of visits per product, purchases, abandonment of carts, among others. User experience improvements have been the subject of several researches ranging from techniques such as eye-tracking (Wong et al., 2014), recommendation systems (Shaikh et al., 2017), and chatbots (Vegesna et al., 2018). Within this context, Knowledge Graphs (KGs) have been adopted as means for knowledge representation.

KGs require constant updates due to their evolutionary characteristics because usually the knowledge they represent evolves over time. In biomedicine-related KGs, for instance, new mutations are added, and existing drugs are modified, following the evolutionary character of the domain. In the e-commerce context, the target of this investigation, significant changes occur over time. For example, products change their availability, as well as how they are compatible one with the other, among other factors. On this basis, the current scenario claims for solutions that provide KGs as credible as possible within e-commerce platforms.

In this context, populating KGs with reliable and error-free information is an arduous task. Domain experts' performing it manually generates inconsistencies and takes a long time. Available solutions to address this challenge are for instance methods for knowledge base completion, knowledge base population, and ontology learning (Asgari-Bidhendi et al., 2021) (Ao et al., 2021) (Gangemi et al., 2017). Knowledge base completion aims to create facts in KGs from knowledge already present in it (Kadlec et al., 2017) (Shi and Weninger, 2018). Knowledge base population and ontology learning from texts are methods that aim to assimilate knowledge from natural language texts to add to existing KGs. This technique is valid to identify the components that are presented in a text to transform it into triples, such as entities, actions, and stopwords.

For this purpose, numerous Natural Language

[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.
QART: A Framework to Transform Natural Language Questions and Answers into RDF Triples.
DOI: 10.5220/0011529200003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 2: KEOD*, pages 55-65
ISBN: 978-989-758-614-9; ISSN: 2184-3228

Processing (NLP) techniques were used and combined to generate text-to-triple transformations as similar as possible to the text author's intention (Liu et al., 2018). In e-commerce, such a transformation can be used in the context of helping decision-making by the customer, influencing him to buy the product or not. The problem of automatically transforming users' NL text from e-commerce platforms into legitimate RDF (Resource Description Framework) triple knowledge is a challenge.

In this article, we propose the development and evaluation of the QART (Question and Answer to RDF Triples Framework), a framework to produce structured representations (RDF triples) from NL texts that express facts in an e-commerce context. More specifically, our goal is to build RDF triples and populate a KG based on NL written user texts. In our approach, the construction of the RDF triples is based on previous answered NL questions made by e-commerce customers about sold products. New coming questions from users explore the knowledge stored in our KG to answer facts about the product, such as compatibility and product specification.

The framework is organized in three steps, composed of 1) Automatic extraction of intents and entities relevant to the e-commerce context; 2) Transformation of the original text into a summarized text, without abbreviations, with shorter and direct sentences. They must be suitable to be transformed into triples in the next step; 3) Generating the triples from the summarized text and adding them to an existing KG in place. In step 2 we explored the concept of templates for the text summarization. We found that the templates can be very useful for our context. They can also be relevant as input examples for the refinement of language models to be useful and application to the step 2.

The remaining of this article is organized as follows: Section 2 discusses the related work. Section 3 presents our proposed framework, its formalization and an application scenario with a practical example. Section 4 reports on the evaluation performed to assess a key step of the framework. Section 5 discusses the open challenges to still advance in the development of QART by pointing out open research directions. Section 6 draws conclusion remarks.

2 RELATED WORK

The act of reading, understanding, and formally structuring knowledge from texts written in natural language has been a recurring challenge at the intersection between computing and linguistics. The advent of formal structures to represent this knowledge with semantic rigor, such as ontologies, triggered the development of numerous tools for constructing ontologies and RDF triples.

The first of these technologies that is worth mentioning is called FRED (Gangemi et al., 2017). FRED uses various NLP tasks to transform multilingual texts into large graphs composed of OWL (Ontology Web Language) and RDF specifications. Among these tasks, we can mention: Named Entity Recognition (NER), where relevant parts of the text that can become resources of the resulting graph are identified; Entity Linking (EL) to connect existing resources of the graph with existing resources in more extensive and more well-known graphs in the community, such as DBpedia; Discourse Representation Structures (DRSs), a first-order logic language for the initial representation of processed text; among other tasks.

The Seq2RDF (Liu et al., 2018) proposes a machine learning model to generate RDF triples from texts using DBpedia as a training base. This model learns to form triples using the encoder-decoder architecture of neural networks. Unlike FRED (Gangemi et al., 2017), this tool does not add several text processing techniques, opting for the approach of training a sequence-to-sequence model. The model cannot generate multiple triples per sentence.

Martinez-Rodriguez *et al.* (Martinez-Rodriguez et al., 2019) proposed a methodology to generate triples from any unstructured text, not only natural language ones. This methodology is composed of crucial steps, similar to the FRED framework. The first step, focused on using the renowned Stanford CoreNLP tool, is feature extraction. Text words are tokenized and segmented to prevent compound words from being separated. The challenge encountered in this step, according to the authors, is the correct identification of errors in sentences that contain grammatical and spacing errors. This is a challenge that we deal with when processing e-commerce texts using QART (our proposal). The second step, called entity extraction, refers to identifying text entities by associating them in large datasets, such as DBpedia. At this point, the word "Barack Obama" from a text on international politics is associated with the Barack Obama resource from DBpedia. After the entity extraction action, the tool goes through the relation extraction step, identifying predicates of triples through a tool called OpenIE. The challenge encountered in this step is that not all types of rules and standards are registered. Finally, the representation step generates the triple RDF. The limitation of the tool proposed by Martinez-Rodriguez *et al.* is related to the fact that it only addresses named entities in the object, which excludes the creation of RDF triples that have literals in the object.

Rossanez and dos Reis (Rossanez and dos Reis, 2019) created a semi-automatic tool that builds Knowledge Graphs from texts of a specific domain: Alzheimer's disease. The texts contain scientific knowledge about the disease. Each sentence is simplified, removing repetition, redundancy, and abbreviations. The tool extracts all triples from the sentences using the Semantic Role Labeling (SRE) technique. The concepts of the generated triples are linked to a public domain ontology of the Alzheimer's Disease domain. The QART framework also builds a KG and maps it with an existing ontology. The QART framework also generates triples using SRE in the triplifying process. However, we combine the templates and transformers instead of SRE in the process of summarizing the text.

Our framework also builds a KG and maps it with an existing ontology, such as Rossanez and dos Reis (Rossanez and dos Reis, 2019). It also creates multiple triples from the text, like FRED (Gangemi et al., 2017) and Rossanez and dos Reis (Rossanez and dos Reis, 2019). Seq2RDF (Liu et al., 2018) also served as inspiration by using neural network models, as well as our framework. However, we could not find in the literature another methodology that focuses on the e-commerce domain and trains the neural network using templates. Our solution combines the templates and transformers instead of Semantic Role Labeling to generate the triples from the text. To the best of our knowledge, there is no evidence of a proposal to transform natural language text into RDF triples in a Q&A e-commerce context.

3 FRAMEWORK QART

This section presents QART, a framework for transformating a set of natural language written texts into RDF triples. Our framework receives as input a set of e-commerce questions and answers $D = \{d_1, d_2, ..., d_n\}$ and outputs a set $T = \{t_1, t_2, ..., t_n\}$ of triples related to D. Triples from T are added to an existing Knowledge Graph (KG). A KG is a directed graph with nodes representing real-world entities such as "Statue of Liberty"; and edges representing relations between entities. A RDF triple (t) refers to a data entity composed of subject (s), predicate (p) and object (o) defined as $t = (s, p, o)$.

Figure 1 presents our methodology responsible for turning natural language texts into RDF triples. Our proposal is organized into three parts: the main flow with the steps, represented by the boxes with the letters from A to C, at the middle of the Figure. In the following, we present details of each step in the subsequent Subsections. Subsection 3.1 presents how the processing and field selection occurs; Subsection 3.2 demonstrates how we employ text-text transformations; Subsection 3.3 stands for our RDF triplying method from the summarized text. All steps expressed in our framework are encoded in Algorithm 1. Along with the presentation of the specific steps, we link it with the corresponding lines in Algorithm 1.

For presentation purposes and clarification of our methods, we provide a running example of a triple created by processing the text in natural language retrieved from an e-commerce Q&A. In particular, the used NL text relates to a user question about the compatibility between a product sold by a store (p_1) and an item possessed by the consumer (ci). From now on, we name it the "consumer item". Figure 2 shows an instantiated version of Figure 1, describing how QART generates a triple from the questions and the answers asked about a product (in the example, we explore the product "Motorcycle Battery 5 ah").

3.1 Step A: Feature Selection and Pre-processing

We consider as input a dataset (v_0) containing all stored actions made in an e-commerce environment by the customers, such as purchases, account creations, product evaluation, questions, answers (blue rectangle of Figure 2). Due to space limitation, Figure 2 only presents part of the whole v_0 dataset. We consider here, for instance, a pair of question and answering in NL text such as question: "Does this motorcycle battery fit on my CG 150 Titan KS?" along with the answer "Yes, the products are compatible" and the purchase of the motorcycle battery made by the customer (original text translated to English language by the authors).

Using the content of dataset v_0 as input, Algorithm 1 asks the user for the dataset fields to be used to construct the triples (line 2 of Algorithm 1). The fields and their contents are pre-processed (line 3 of Algorithm 1). The pre-processing procedure changes or removes noisy data from the input text. The noisy data are words that do not add meaning to the text or the triple construction in further steps. Examples of noisy data are stop-words, abbreviations, greetings, and punctuation.

The algorithm 1 identifies entities and intents from fields of v_0. Intents are the types of action described in one or more sentences. In an e-commerce context,

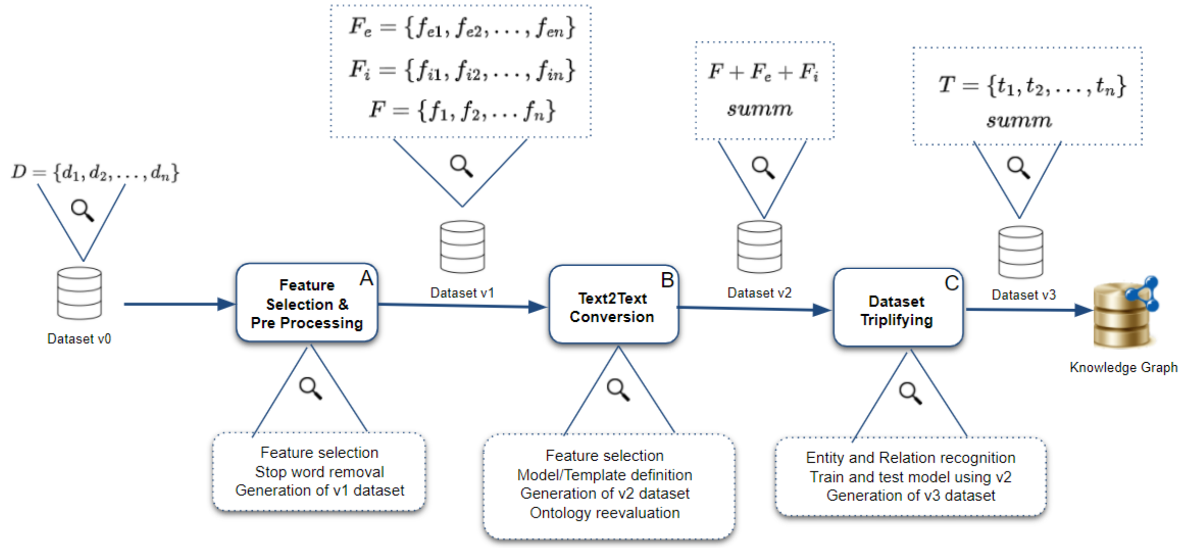

Figure 1: QART Framework composed of three steps (rectangles in the middle of figure), a description of each step (rectangles in the bottom of figure) and the content of each input/output dataset (rectangles in the top of figure).

there are user sentences that refer to purchase intent (”I would like to buy two units of this shoe”), product availability intent (”Do you have it in blue?”), and shipping intent (”What is the shipping value to Rio de Janeiro?”). Identifying intents is relevant to determine which type of triples t are generated and stored in the KG in step C of QART (*e.g.*, purchase, availability). Entities are information in the sentences on which intents act. In an e-commerce context, it is possible to identify entities related to product specifications, such as voltage, size, model, weight, and year of manufacture, among others. The QART framework uses entities as resources of the triples in step C (s and o of each t). The set of all chosen fields $F = \{f_1, f_2, ...f_n\}$, their respective entities $F_e = \{f_{e1}, f_{e2}, ..., f_{en}\}$ and intents $F_i = \{f_{i1}, f_{i2}, ..., f_{in}\}$ forms the dataset v_1.

Figure 2 shows a running example used throughout this section. Among all fields of the v_0 dataset, the user - an ontology maintainer, for instance - defines that the fields containing the title, the question, and the answer are essential for the triple generation task ($F = \{”ProductName”, ”Question”, ”Answer”\}$) (line 2 of Algorithm 1). Figure 2 (green rectangle) shows these three attributes related to the “*Motorcycle Battery 5 ah*” product (p_1). From these three attributes, the framework proceeds to the task of finding important parts in the text: the entities and intents. Figure 2 presents that the identified intent of the text is “compatibility” and the entities are “car name”, “model”, and “year”. Together with the three original fields (“Product Name”, “Question”, “Answer”), these fields form the v_1 dataset (green rectangle).

Among the types of intents found in the questions asked by the client, we grouped them into two distinct classes: stable and mutable. Answers from questions with stable intents tend not to vary or vary little over time. As an example, in the e-commerce domain, we have the intent of product specification and compatibility. Both can undergo some modifications. Responses with changing intents vary more frequently, ranging from minutes to months. As an example, we have the intents of product availability and shipping options. A particular product may be available now, but it may no longer be available within minutes. Such categorization is essential to better define the scope of intents in which the resulting KG will keep knowledge. Questions with stable intent result in stable triples in the resulting KG. Questions with changing intentions result in triples and KGs that vary over time and require a different approach to study and implementation, based mainly on Temporal Knowledge Graphs (Rossanez et al., 2020).

The QART deals with triples arising from stable intents, and this categorization is performed in Step A of the framework.

3.2 Step B: Text2Text Conversion

The dataset v_1 serves as an input for Step B (second rectangle in the center of Figure 1). QART summarizes the most suitable fields (v_1), generating a short and meaningful text that expresses the semantics of the questions and answers (v_2). The rationale is that generating triples from summarized, condensed, and factual texts can be something positive, facilitating the generation of triples in Step C.

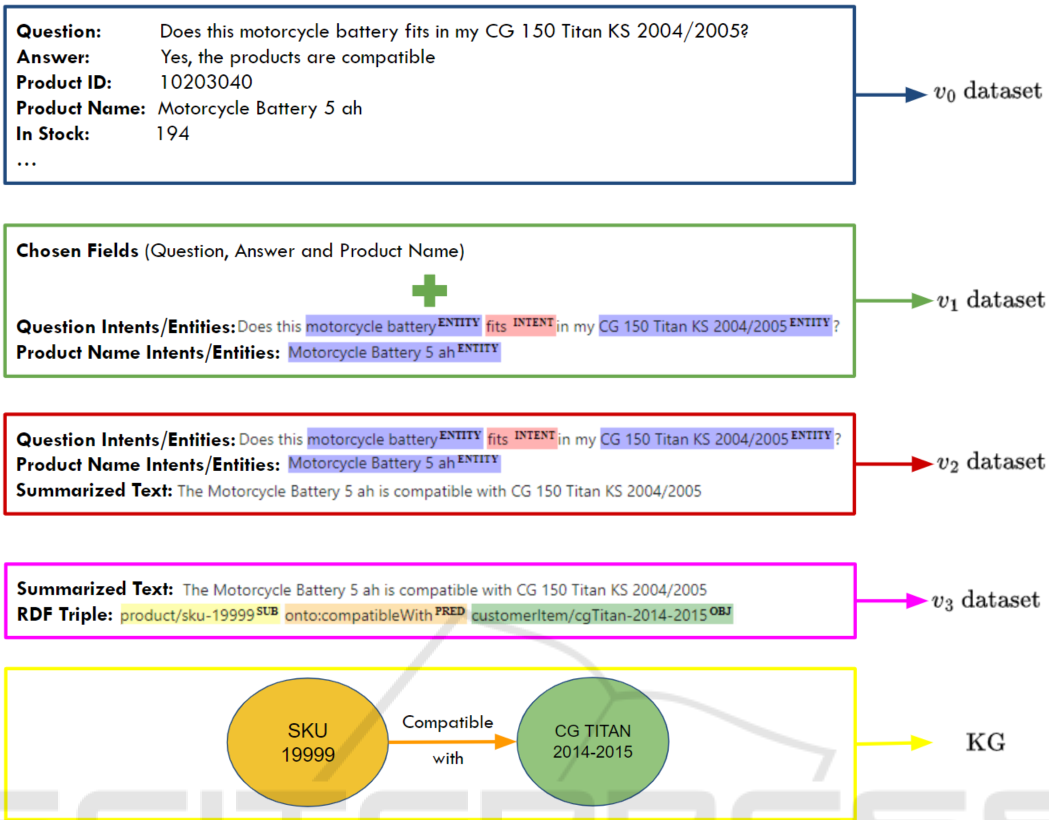

Figure 2: Example of QART functioning examplified with the use of one product and one related question and answer. The blue rectangle illustrates the v_0 dataset; the green, red and pink illustrates v_1 to v_3, respectively. The yellow rectangle indicates the resulting triple t that is added to the existing KG.

To perform this text-to-text conversion, line 7 of Algorithm 1 uses the selected fields from dataset v_1 and transforms them into a single condensed field *summ*. The v_2 dataset has two fields: the full text that contains questions, answers, intents, and entities collected in the v_1 dataset and the second field, which is a summarized text *summ* (line 8 of Algorithm 1).

To summarize the text, the QART framework uses templates. The use of templates is based on filling the summarized texts *summ* with entities F_e. Choosing the appropriate template for each *summ* is based on the F_i intent found in the question/answer pair.

Using the example from Figure 2, the summarized column contains text capable of briefly expressing the content of the columns referring to the title, question, and answer of the motorcycle battery product (field "Summarized Text", red rectangle of Figure 2). For this text-to-text transformation, the framework generates *summ* "The Motorcycle Battery 5 ah is compatible with CG 150 Titan KS 2004/2005", summarizing that the product p_1 entitled "Motorcycle Battery 5 ah" is compatible with the customer item *ci* "CG" model "Titan KS" with the year "2004/2005". The v_2 dataset is filled with all these fields (and their intents) and *summ*.

In summary, from the v_1 dataset, we have a v_2 filled with summarized text suitable to be transformed into a triple in the next step. Step B is where the original text is transformed into a "factual text", easier to be added to a KG due to its structure than the natural language question text without any kind of pretreatment.

3.3 Step C: Text Triplifying

The dataset v_2 serves as an input to Step C, which is the dataset triplifying. This transforms the summarized facts *summ*, into triples t (line 9 of Algorithm 1). The fact *summ* must contain statements with intent of the stable type, described in Section 3.1.

The triplifying task is done using the Semantic Role Labeling technique (Màrquez et al., 2008), which identifies subject, predicate, and object of the summarized facts *summ*. This generates the dataset v_3. Triples, part of the v_3 dataset, are returned by the algorithm to be added in the KG.

Table 1: List of all templates used in the evaluation. The first column represents the answer intent. The second column shows the templates, four for each type of answer intent.

Intent	Template
fits	The product [PROD] is compatible with [BRAND] [MODEL] [MODEL_SPEC] [YEAR]. This product [PROD] fits in [BRAND] [MODEL] [MODEL_SPEC] [YEAR]. The car [BRAND] [MODEL] [MODEL_SPEC] [YEAR] is compatible with [PROD]. The product [PROD] is suitable for use in [BRAND] [MODEL] [MODEL_SPEC] [YEAR].
does not fit	The product [PROD] is incompatible with [BRAND] [MODEL] [MODEL_SPEC] [YEAR]. The product [PROD] does not fit in [BRAND] [MODEL] [MODEL_SPEC] [YEAR]. The car [BRAND] [MODEL] [MODEL_SPEC] [YEAR] is incompatible with [PROD]. The product [PROD] is not suitable for use in [BRAND] [MODEL] [MODEL_SPEC] [YEAR].

The *summ* field is key to step C, where QART performs a text-to-triple transformation process, transforming *summ* "The Motorcycle Battery 5 ah is compatible with CG 150 Titan KS 2004/2005" into an RDF triple t. Both *summ* and t are represented in Figure 2 inside the pink rectangle, with field name "Summarized Text" and "RDF Triple", respectively. Together they form the v_3 dataset. Additional discussions on text-to-triples transformation can be found at Section 5.

In the yellow rectangle of Figure 2, we identify a representation of an existing Knowledge Graph containing the newly created triple in Step C. It is possible to identify that the product p_1 (the battery) is the subject of the RDF triple (in yellow), the intent f_{i1} is the predicate (in orange) p and the consumer item ci (the motorcycle) is the object o.

When a new question is asked about the compatibility between the motorcycle p_1 and the battery c_i, the KG is ready to answer it, not requiring any additional human intervention to answer.

3.4 Implementation Aspects

For Step A, the framework uses RASA (Sharma and Joshi, 2020), a conversational AI platform to identify the intents (F_i) and entities (F_e) of the chosen attributes after pre-processing (green rectangle in Figure 2). Through the use of a word embeddings model, RASA identifies the intention expressed by a given input phrase. In this case, the question asked by the client. To identify entities, we use models based on Conditional Random Fields (Lafferty et al., 2001).

For Step B, the generation of *summ* is achieved using templates. Table 1 shows some template examples. All templates use the same entities, and there is more than one template for each intent. The choice to create more than one template per intent is motivated by introducing linguistic variability in step B. We understand that the more diverse the templates are in linguistic terms, the lower the chance of bias. Templates can be used to fine-tune text generator templates (such as GPT 2 (Radford et al., 2019) or T5 (Raffel et al., 2020)), create a large volume of data, and form artificial datasets. As seen in Table 1, an intent is capable of generating numerous summarized texts with linguistic variability.

For Step C, QART processes *sum* using Semantic Role Labeling from IBM Watson (Ferrucci, 2012) and the resulting triples are in n-triples format[4].

Algorithm 1: Transforming natural language text from a set of question and answers about products into RDF triples.

Require: D
$F \leftarrow chooseFields(D)$
$F \leftarrow preprocessFields(F)$
$F_e \leftarrow identifyEntities(F)$
$F_i \leftarrow identifyIntents(F)$
if $F \neq \emptyset$ **then**
  $v_1 \leftarrow F \cup F_e \cup F_i$
  $summ \leftarrow getSummary(v_1)$
  $v_2 \leftarrow mergeColumns(v_1) \cup summ$
  $v_3 \leftarrow triplifyDataset(v_2)$
  if $v_3 \neq \emptyset$ **then**
    return $v_3.t$
  end if
end if

4 EVALUATION

This section evaluates step B of the QART framework, responsible for transforming a text composed of the question and answer into a summarized text. We measure the quality of the sentences generated using templates based on the number of correctly identified intents and entities. We compare the entities and intents identified by a set of evaluators with the entities and intents found by QART in a set of real compatibility questions and answers in the context of a Brazilian e-commerce platform. The quality of these

[4]https://www.w3.org/TR/n-triples

templates directly affects the quality of the summarized texts and, consequently, the triples generated in the last step of the framework. The summarized texts present stable facts (section 3.1) that are triplified in the following steps of the framework. In this evaluation, we investigate compatibility Q&A, one of the stable type examples.

4.1 Setup and Procedures

As our first step, we created a dataset that contains questions asked about the products by the customer and answers from the attendant. Through step 2 of the QART framework we generated a set of summarized texts using templates. The summarized text should succinctly express the question's intent, the entities involved, and the answer's intent. For example, in a question with compatibility intent, the answer must contain an affirmative or negative answer regarding the connection between the consumer's item *ci* and the product p_1, thus revealing its intent.

We retrieved 3737 questions of different types of intent from the ten largest stores from the marketplace platform with the highest flow of questions and answers. These questions were asked between January and February 2022. Of the 3737, we randomly chose 20 questions and answers from each of the ten stores, totaling 200 real and random examples of questions asked in e-commerces that are GoBots customers. The dataset containing these 200 examples should contain examples whose question intent was of the compatibility type, which is the focus of this experimental evaluation. We further discuss implications of addressing other types of intents in Section 5.

Parallel to the population of this dataset, we created a dataset of templates (cf. Table 1). Each of the 200 sentences from the evaluation dataset, combined with one of the randomly chosen templates described in Table 1, generated a summarized sentence. Each template has a type of response intent (column 1 in Table 1) and its respective content (column 2 in Table 1). The content in square brackets illustrates where each entity is inserted in the template to generate the summarized text.

There are four templates for each response intent type, resulting in a total of eight templates. The intention "fits" refers to affirmative responses regarding the compatibility between the consumer's item and the product; and "does not fit" refers to products and consumer items that are not compatible. After identifying the response intent, one of the four templates available for that intent is randomly chosen to generate the summarized phrase. The dataset with 200 summarized texts is generated by processing the dataset used in this evaluation containing 200 compatibility questions and answers with the dataset containing the eight templates.

Figure 3 shows two examples of summarized texts and fields used for the evaluation. The example (A in Figure 3) states that the consumer item fits the product (intent). Each entity of the consumer item is identified by different colors. The example (B in Figure 3) presents a "does not fit" intent, two entities from the consumer item (model and year) and two missing entities in the summarized text. The v2 evaluation dataset generated by QART contains the following data:

- 200 rows of summarized texts, based on the combination of product title, question and answer with templates;
- 200 rows of response intents (red tag in Figure 3), which is automatically filled with "fits" and "does not fit" values;
- 41 filled rows of automobile brands (yellow tag in Figure 3) found in the summarized text;
- 187 rows filled with automobile models (pink tag in Figure 3) found in the summarized text;
- 162 filled rows of the car's manufacture year (purple tag in Figure 3) found in the summarized text;

Each of the entities has values less than 200 due to the QART framework not being able to find such entities in the summarized text; observe the missing entities from the example B in Figure 3. It is part of this assessment to determine how many of these missing fields were erroneously unidentified; and fields that were identified as a particular entity but belonged to a different entity classification.

Gold Standard. This evaluation was only possible due to the comparison against a gold standard dataset created a priori. The gold standard was composed of 200 sets of questions and answers, 200 intent classification ("fits", "does not fit", "not a compatibility question") and 508 annotated entities ("brand", "model", and "year"). We asked a total of six independent evaluators to identify the intent of the answer and the four different entities in each pair. All of the evaluators had background experience in Artificial Intelligence and each of them analysed a random number of intents and entities.

4.2 Results

We present the results of the QART's accuracy, precision, and recall compared to the gold standard. Table

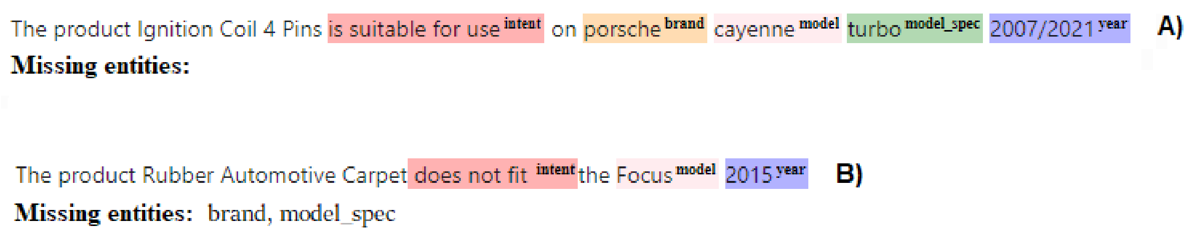

Figure 3: Two examples of summarized texts and their intents and entities.

Table 2: Results of the comparison between intents in gold standard and predicted values from QART.

		Gold Standard			
		fits	**does not fit**	**no compatibility**	**Total**
QART	**fits**	135	7	5	147
	does not fit	4	44	5	53
	Total	139	51	10	200

2 shows the results achieved in the intent classification; and Table 3 shows the results achieved in the entity discovery.

In Table 2, each cell displays the sum of the intersection between the results obtained by the QART - split by the columns "fits" and "does not fit" - and the annotated results in the formation of the gold standard - split by the lines "fits", "does not fit" and "is not a compatibility question". We observe that the framework was correct in case the consumer item fits the product in 135 cases; and 44 cases in which the framework correctly detected that *ci* and *p* were incompatible (i.e., "does not fit"). Such cases, identified as True Positive and True Negative, show that our framework obtained 179 out of 200 compatibility intentions (89.5%).

Among the 10.5% of erroneously categorized cases, according to Table 2, we have 4 cases (2%) in which the QART incorrectly categorized the intention of the response as "does not fit", classified as False Negative cases. The opposite situation occurs in another 7 cases (3.5%), erroneously categorizing intent as "fits", classified as False Positive cases.

The third column of Table 2 shows the remaining of the cases with classification error by the framework (5%), which erroneously classified ten questions with other intentions (thanks, shipping, availability) as compatibility questions. Equation 1 and Equation 2 show the weighted precision and recall based on the results described in Table 2.

The members of Equation 1 are, respectively, the precision of "fits" cases (P_F), the precision of "does not fit" (P_{NF}) and the precision of "not compatibility question" (P_{NC}). The recall is calculated by Equation 2 with three weighted recalls (R_F,R_{NF}, and R_{NC}). The weighted average is used instead of a simple average because the three classes (F, NF, and NC) are not balanced.

$$Precision_W = P_F + P_{NF} + P_{NC} = 85.00\% \tag{1}$$

$$Recall_W = R_F + R_{NF} + R_{NC} = 89.5\% \tag{2}$$

Table 3 presents the accuracy results in evaluating each of the three entities identified by the QART. Equation 3 shows that 516 entities (86%) out of 600 (200 brands, 200 models, and 200 years) were correctly identified by QART. Each of the 200 summarized texts generated 0 to 3 entities. Among the 200, there were 4 cases (2%) in which no entity was correctly identified; 11 cases (5.5%) in which only one entity was correctly identified; 50 cases (25%) in which two entities were correctly identified; and 135 cases (67.5%) where all three entities were correctly identified.

$$Accuracy = \frac{TP + TN}{AllOccurrences} = \frac{516}{600} = 86\% \tag{3}$$

Table 3: Results of the comparison between 3 entities in gold standard and predicted values from QART.

	Brand	**Model**	**Year**	**Total**
Total	183	168	165	516
%	91,5	84,0	82,5	86,0

Figure 4 shows two real-world examples of triples generated from the *summ* text of this evaluation.

4.3 Discussion

We understand that high precision and recall in intent results and high accuracy in entity results indicate good quality in the generated templates. The first observation concerns the correct identification of the question intent: only 5% of these were erroneously

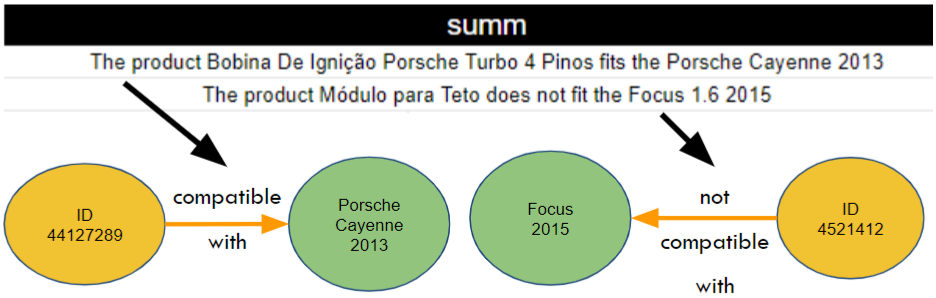

Figure 4: Example of two triples created from *summ* texts. The first triple shows a compatible product and the second triple an incompatible product.

categorized as compatibility questions. The second finding concerns the binary classification of the response intent, which classifies them as "fits" or "does not fit". In addition to finding out that the question is about compatibility, the framework proves to identify whether the answer about compatibility is positive or not in 85% of cases.

The third observation concerns the entities. The entity with the highest accuracy was brand, with 91.5%, whereas the one with the lowest accuracy was the year, with 82.5%. We believe that this result was due to the nature of each of the fields in question: the brand is a field containing a simple text, often not mentioned in the question/answer texts; the year is a field subject to numerous difficulties. The same year can be represented with different numbers of digits ("1994" and "94"), separated by different types of characters ("1994/1995", "1994-1995"), containing ranges ("1994 to 1998"), be confused by other car features ("1.8", "180 horsepower"). Such syntactic and semantic differences make identification difficult by the QART.

The fourth finding refers to the positive result of the accuracy of 86% of correctly identified entities and 92.5% of cases with two or more correctly identified entities. Some products and cars have very specific compatibilities, serving only a particular model, brand, and year. Thus, the more the proposal can identify all the entities of a compatibility phrase, the more assertive its answer can be. Identifying no entity in the question and answer makes the correct answer impossible. There is no way to detect that the product is compatible with a consumer item without specificity.

5 CHALLENGES

This section presents the main challenges involved in building the QART framework.

Text Interpretation in Natural Language. The first significant challenge refers to the difficulties encountered in reading and identifying terms. This challenge is present in all three stages of the framework (Figure 1), caused by inherent characteristics of in natural language texts, such as the presence of abbreviations, colloquialisms, and regionalisms.

Step A is negatively impacted mainly by grammatical errors because they can interfere in the identification of subjects and objects of the sentences (entities) and the identification of the types of actions (intentions). In the example in Section 3, if the model or year of the motorcycle had been written with some grammatical error (e.g. "Titan SK" and "204" instead of "2004"), the following steps would probably be affected.

In the following steps of the framework, where there is text-to-text and text-to-triple transformation, the difficulty in processing texts with ambiguities, irony, and sarcasm is worth noting. If the text referring to the question in Section 3 was related to product criticism, instead of a compatibility question using irony and sarcasm, there would be an adverse effect on the pipeline.

Application of Language Models For Text-to-text Transformation. In step B, we have performed evaluations with templates and machine learning models to generate the text transformations to build RDF triples in step C. The ideal NL text transformation results in a short, synthetic text that can be easily transformed into a subject-predicate-object format, unambiguous, and well-defined semantics.

Nowadays, Transformers (Vaswani et al., 2017) are considered the state-of-the-art for several machine learning-based NLP tasks. We understand that to transform the text of the questions and answers into a shorter (summarized) text requires models previously trained with a dataset related to text summarization or text-to-text transformation. Among the summarization models trained with large data, we can mention

Pegasus (Zhang et al., 2020), BART (Lewis et al., 2020), and T5 (Raffel et al., 2020). To the best of our knowledge, there is no investigation exploring transformers that can generate triples from summarized Q&A sentences. We observe it as a key challenge and it is in the roadmap of our future investigations.

Portuguese Language. We used NL questions, answers, and product titles registered on large Brazilian e-commerce platforms. Step B (cf. Figure 1) of our framework requires that the text is converted to a summarized text. To this end, a future research venue refers to the use of pre-trained models in Portuguese language; or an additional transfer learning step to understand the Portuguese text with a model pre-trained in English corpora. There are some alternative multilingual models available for investigation, such as mT5 (Xue et al., 2021), a multilingual version of T5, which was trained with datasets that include the Portuguese language. This solution can be explored if we opt to use pre-trained transformer models.

Structure of Existing KG. Step C generates the v_3 dataset that contains the summarized text *summ* and the triple t associated with it. This triple should be inserted to an existing KG to add more knowledge to it. For this purpose, it is necessary to guarantee that the triple, an instance, is compatible with the predefined ontological elements, the class. Figure 5 illustrates what the existing KG structure should be for the running example of Section 3. In red, we have the classes and, in green, the instances of these classes. A triple with compatibility intent generated by the QART must be conform to this knowledge representation. For the example of Section 3, three classes are required to represent the product p_1, the consumer item ci, and the compatibility between them. Texts and triples containing intents different from "compatibility" must have another class structure.

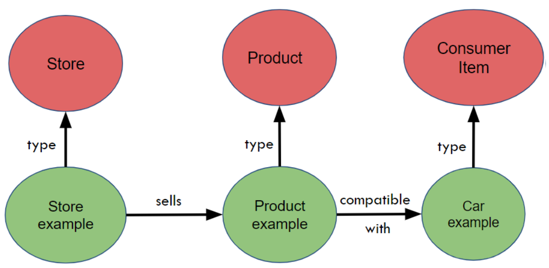

Figure 5: Synthetic representation of a KG that is prepared to store knowledge about compatibility between $prod_1$ and c_i. The red circles refer to the classes Store, Product and Consumer Item. The green circles refers to examples of instances from the 3 classes.

6 CONCLUSION

Using natural language texts to automatically discover meaningful data and fill semantic-enhanced structures, such as KG, is a promising task. Much data is lost for not being mined, such as questions and answers about products in an e-commerce context. This investigation proposed the QART framework to generate RDF triples with a pipeline composed of entity and intent detection, text-to-text transformation, and text-to-triples generation. We described the framework and the challenges in its further development. Our study provided an illustrative example to describe the potentialities of our solution to generate RDF triples. In particular, we evaluated the use of templates for text summarization as a key step in our solution. We found that they can be very useful as training data for machine learning models, given the high accuracy, precision, and recall achieved. Future work involves the development of an interactive software tool that guides the users throughout the process, such as a data engineer who fills a KG with relevant data based on our framework.

ACKNOWLEDGEMENTS

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

REFERENCES

Ao, J., Dinakaran, S., Yang, H., Wright, D., and Chirkova, R. (2021). Trustworthy knowledge graph population from texts for domain query answering. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 4590–4599. IEEE.

Asgari-Bidhendi, M., Janfada, B., and Minaei-Bidgoli, B. (2021). Farsbase-kbp: A knowledge base population system for the persian knowledge graph. *Journal of Web Semantics*, 68:100638.

Ferrucci, D. A. (2012). Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15.

Gangemi, A., Presutti, V., Reforgiato Recupero, D., Nuzzolese, A. G., Draicchio, F., and Mongiovì, M. (2017). Semantic web machine reading with fred. *Semantic Web*, 8(6):873–893.

Kadlec, R., Bajgar, O., and Kleindienst, J. (2017). Knowledge base completion: Baselines strike back. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 69–74.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Liu, Y., Zhang, T., Liang, Z., Ji, H., and McGuinness, D. L. (2018). Seq2rdf: An end-to-end application for deriving triples from natural language text. In *CEUR Workshop Proceedings*, volume 2180. CEUR-WS.

Màrquez, L., Carreras, X., Litkowski, K. C., and Stevenson, S. (2008). Semantic role labeling: an introduction to the special issue.

Martinez-Rodriguez, J. L., Lopez-Arevalo, I., Rios-Alvarado, A. B., Hernandez, J., and Aldana-Bobadilla, E. (2019). Extraction of rdf statements from text. In *Iberoamerican Knowledge Graphs and Semantic Web Conference*, pages 87–101. Springer.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Rossanez, A. and dos Reis, J. C. (2019). Generating knowledge graphs from scientific literature of degenerative diseases. In *SEPDA@ ISWC*, pages 12–23.

Rossanez, A., Reis, J., and Torres, R. D. S. (2020). Representing scientific literature evolution via temporal knowledge graphs. CEUR Workshop Proceedings.

Shaikh, S., Rathi, S., and Janrao, P. (2017). Recommendation system in e-commerce websites: A graph based approached. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 931–934. IEEE.

Sharma, R. K. and Joshi, M. (2020). An analytical study and review of open source chatbot framework, rasa. *International Journal of Engineering Research*, 9.

Shi, B. and Weninger, T. (2018). Open-world knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vegesna, A., Jain, P., and Porwal, D. (2018). Ontology based chatbot (for e-commerce website). *International Journal of Computer Applications*, 179(14):51–55.

Wong, W., Bartels, M., and Chrobot, N. (2014). Practical eye tracking of the ecommerce website user experience. In *International Conference on Universal Access in Human-Computer Interaction*, pages 109–118. Springer.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.