MediBot: An Ontology-Based Chatbot to Retrieve Drug Information and Compare its Prices

Caio Viktor S. Avila[1], Wellington Franco[1], Amanda D. P. Venceslau[1], Tulio Vidal Rolim[1], Vania M. P. Vidal[1] and Valéria M. Pequeno[2]

[1] Department of Computing, Federal University of Ceará, Fortaleza-CE, Brazil
caioviktor@alu.ufc.br, wellington@crateus.ufc.br, amanda.pires@ufc.br,tuliovr@alu.ufc.br, vvidal@lia.ufc.br
[2] Universidade Autónoma de Lisboa, Departamento de Engenharia e Ciências da Computação e Autónoma TechLab, Portugal
vpequeno@autonoma.pt

Abstract. In this article, we present the MediBot. MediBot is a chatbot for querying drugs information. The presented system acted as a single access point for natural and simplified information retrieval of drugs, prices, and its risks. The chatbot has two modes of operation: Quick Response and Interactive modes. The first answers questions asked in natural language, while the second has three interactive tasks, namely Browser, Query, and Price Comparison. We present here the system architecture, the Linked Data Mashup's construction process, and Chatbot MediBot's activities modes, focusing on the new Price Comparison's task. This task presents the best prices for medicines and their best potential substitutes extracted in real-time from the Web with the help of the information obtained from a linked data mashup.

Categories and Subject Descriptors: H.2 [**Database Management**]: Database Applications; H.3 [**Information Storage and Retrieval**]: Systems and Software; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

Keywords: Chatbots, Data Integration Systems, Semantic Web, Medical Informatics, Drugs

1. INTRODUCTION

According to the World Health Organization (WHO), the rational use of drugs by patients occurs when they receive drugs appropriate to their clinical conditions, in doses relevant to their individual needs, for a reasonable period and at the lowest cost to themselves and the community [WHO 1987]. Thus, rational use is an essential prerogative for effective drug use. However, according to [Aquino 2008], the Brazilian population's consumption practices are very far from such a scenario, where a first step to promote the rational use of drugs is through education and information of the community.

In the *Web*, there is a wide range of drug data that can assist in guiding the correct medication. However, such data are made available through technical vocabularies aimed at health professionals, and their comprehension is a challenge for lay users. Another problem is that most of this data is in a proprietary format, such as spreadsheets, relational database backups, or available only through Web pages, making it difficult to extract the information, requiring specific recovery methods for each source [Schyve 2007].

In this article, we seek to develop a method that democratizes access to knowledge about the domain of drugs - and empowering the user with the knowledge to ensure the rational use of drugs by the general population. As a solution to this scenario, we came to perform the semantic integration of

several publicly available data sources on the Web, generating a *Linked Data Mashup* (LDM). As a result, having that enables information retrieval [Hoang et al. 2014] and acts as the main Knowledge Base (KB) to the system. The LDM allows querying multiple datasets through a single access point and a common vocabulary accessible to lay users.

As a query and knowledge access interface, we have developed *MediBot*. *MediBot* is a chatbot for a well-known instant messenger client. The chatbot provides the user access to the information contained in the KB using natural language. Allowing non-technical users to query and retrieve their information of interest without understanding the data structure and query languages. *MediBot* has two modes of operation *Quick Response mode* and *Interactive Mode*. The first answers questions asked in natural language, while the second has three *interactive tasks*, namely *Browser, Query*, and the new activity, *Price Comparison*.

As an innovation of this work, in comparison with in [Avila et al. 2019], we focused on the point of *"...and the lowest cost to themselves and the community."* of the definition of rational use of drugs. According to [G1 2016], the price difference between drugs can exceed 5,000% from one pharmacy to another and between different versions of the same drug. Thus, in this article, we present the *Price Comparison* interactive task. This task compares the prices available for a drug and its similar. To this end, the Web is retrieved in real-time for the prices offered for the various presentations of a drug in the country's leading pharmaceutical chains. Also, as a differentiated, the system seeks not only provides a single drug but also for its similar drugs. To find similar drugs, we use the information contained in our LDM. These are the main contributions of this article:

—We propose a method that democratizes access to knowledge about the domain of drugs;
—We have developed *MediBot*. *MediBot* is a chatbot for a well-known instant messenger client;
—We present the *Price Comparison* interactive task. This task compares the actual prices available for a drug and its similar.

The remainder of this article is structured as follows: Section 2 presents background. In section 3, the system architecture is presented. Section 4 describes the process of construction of the Linked Data Mashup produced in the project. Section 5 gives an overview of *MediBot*. Section 6 presents the performed assessments of the system. Finally, section 7 presents the conclusions and final considerations of the authors - besides, section 8 present appendices figures.

2. BACKGROUND

This section presents a brief overview of the topics involved in the development of this work. Section 2.1 presents the main concepts involved in the context of Semantic technologies. In turn, Section 2.2 addresses the technologies involved in the development of Natural Language Interfaces.

2.1 Semantic Technologies

The Semantic Web is an extension of the traditional Web (Document Web), seeking to allow humans and computers to work in cooperation[1]. The Semantic Web is an initiative to add semantics and machine-understandable meaning to information on the traditional Web being proposed by the founder of the World Wide Web (WWW), Tim Berners-Lee [Berners-Lee et al. 2001].

The main difference between the Document Web and the Semantic Web is that while in the first, the information is stored in the format of Hypertext Markup Language (HTML) pages, the second, the data is structured in Resource Description Framework (RDF) graphs structured according to a taxonomic scheme, called Ontology [Berners-Lee et al. 2001]. In turn, the term Linked Data (LD)

[1]https://www.w3.org/html/

refers to the set of best practices for publishing and connecting structured data to the Data Web [Bizer et al. 2011]. The LD technologies are used to create links between data from different sources. These sources can be as diverse as databases maintained by two organizations in other geographic locations, or simply, heterogeneous systems within the same organization that historically have not been interoperable at the data level [Bizer et al. 2011].

Extracting useful information is a challenging task, requiring time, effort, and technical and analytical skills. Semantic Query makes it possible to retrieve both explicit and implicit knowledge. In this type of Query, information is implicitly derived based on syntactic, semantic, and structural information contained in the data. For the inference of new information, axioms are used as rules of inferences. Such axioms are defined in a formal language and describe how further information can be inferred from information already known (*e.g.*, "*if Caio is a man, then Caio is also a person*"). Semantic queries act on KBs, allowing the Query to process the actual relationships between information and infer responses from the data network. This contrasts with the semantic search, which uses semantics (the science of meaning) in unstructured text to produce a better search result.

2.2 Natural Language Interface

Natural Language (NL) is one of the most efficient means of communication and is used by humans, either by spoken language or by text. Bearing this in mind, NL's use in human-machine interaction presents itself as a natural and attractive option. The Natural Language Interface (NLI) is a type of human-machine communication interface realized through sentences and phrases, like those used by humans in day-to-day conversations, that act as commands for computer systems [Hendrix et al. 1978]. Natural Language Processing (NLP) is a sub-field of artificial intelligence that deals with human-machine interaction via natural language. NLP explores how computers can understand and manipulate natural language by text or by voice [Chowdhury 2003].

2.2.1 *Chatbots.* ChatterBot, or simply Chatbot, is a computational agent who can engage in NL dialogues as a human. While some chatbots are designed for the sole purpose of simulating humans, as in banal conversations, there are also goal-oriented ones. Examples of these goals are the automation of tasks previously performed by human attendants, *e.g.*, customer service, online vendors, messaging, and news [Dale 2016].

According to [Radesko 2012], the evolution of chatbots has relied on many technologies, among which, the following can be highlighted: **Pattern Matching**, a more common approach adopted. In this approach, the developer defines an input template expected by the Chatbot and an output template to be answered. The template can be determined using regular expressions (Regex); **Parsing**, in this approach, the original input text is segmented into tokens, which are then structured as a tree representing the input sentence's linguistic structure. This approach can be implemented with the use of grammatical parsing that assess whether the input is grammatically accepted by the language, following the use of terms and their correct order; **Markov Chain Models (Markov Chain)**, this approach sees a conversation as a state machine, where the probability of the next state (chatbot response) depends only on the current state (user question). Statistical Markov models are then used to calculate the likelihood of an exit; **Ontologies or Semantic Networks** uses Ontologies to hierarchically structure knowledge about a given domain, containing its concepts and relationships. It is then attempted to map the user's input to existing terms in Ontology to build an interpretation of the structured question as a graph. This allows the use of graph search techniques for the inference of missing fragments in the question; **AIML**, The Artificial Intelligence Markup Language (AIML) is an XML-based markup language for creating NL conversations for chatbots. AIML uses an approach similar to pattern matching, where an expected input pattern is defined along with its corresponding output pattern. AIML's differential lies in the fact that it is a language of structured patterns (question-answer pattern), which guarantees reuse and interoperability. The great asset of AIML lies on its capacity for self-reference where a pattern can call itself and pass a new entry

as a parameter; **ChatScript**: It aims to be the successor of AIML, incorporating characteristics of ontologies, such as concepts, variables, and facts to the dialogue management mechanism of AIML. Its code is available at GitHub [Radesko and Mladenic 2018]; **RiveScript**: Like AIML and ChatScript, it is a scripting language for creating chatbots. RiveScript has become quite popular in creating chatbots, implementing several languages. Such as Go, Java, JavaScript, Perl, or Python. One of the strengths of RiveScript is its simplicity, being implemented in the plain text following some standards, with no need to use XML [Petherbridge 2019].

3. RELATED WORK

[Jovanovik 2017] shows how to perform the integration and publication of data about drugs of twenty and three countries, however, the Brazil there are not between these. Also, the vocabulary defined for the integration does not present the databases selected in this work.

[Natsiavas et al. 2017] present a model for the use of several databases integrated through *Linked Data*, in particular datasets of the project *BIO2RDF*[2], for the mining of signs of adverse reactions of drugs, showing the potential of linked databases in the drug domain. However, the model was designed to be used as part of a data mining platform, so it does not address how users will access data. This is also the aim of the work by [Nováček et al. 2017].

[Vega-Gorgojo and Slaughter 2016] present *PepeSearch*, a system that facilitates searching between different sources Linked Data in the field of drugs and health, such as *Drug Bank* (DB) and *Sider*. The system provides a faceted query interface that allows the user to search on multiple data sources. However, the system is best suited for use by specialists and researchers, besides it has a powerful yet complex query interface. Moreover, the system does not have data on drug risks in pregnancy, despite having the side-effects data.

MedChatBot is a chatbot for medical students based on the open-source AIML UMLS to generate responses to queries through knowledge extraction [Kazi et al. 2012]. Some approaches focus on building a chatterbot that seeks to alert users about the risk of a drug interaction. The work of [Avila et al. 2019] seeks to do just that.

Finally, some apps for *smartphone* like *Consulta Remédios*[3], *MediPreço*[4], etc. provide drug price comparator service. However, such services only compare prices of the same drug presentation to different retailers (pharmacies and drugstores) and do not directly compare the price of similar drugs. Also, such services require the user to download and install an *application*, which is time-consuming to install and storage space on the device, which can be a disadvantage when the user only needs to perform quick and timely queries. With the use of a chatbot, this difficulty is overcome, as the user only needs send messages to the chatbot via a messaging application (increasingly common in people's lives), with no need to download or install a new application just for this function.

MediBot allows queries to be performed using natural language, transforming them into SPARQL queries on a Linked Data Mashup (Section 5), focusing on price comparison to be consumed by a chatbot, involving drugs data provided by ANVISA and Sider sources. Finally, MediBot presents itself as a reasonable instrument in promoting access to information for the general public. According to a set of conversation flows, this chatbot works to facilitate access to information quickly.

[2]http://bio2rdf.org/
[3]https://play.google.com/store/apps/details?id=com.consultaremedios
[4]https://play.google.com/store/apps/details?id=br.com.ilevel.medipreco

4. SYSTEM ARCHITECTURE

To define our architecture, we use the methodology proposed by [Neto et al. 2013]. In this work, the authors propose an RBA tool (R2RML By Assertion) that automatically generates customized R2RML mappings based on a set of semantic mappings that model the relationship between the relational database schema and a target ontology in RDF. They define a methodology that divides the process into three stages, separating the logical and data parts. We use the same methodological inspiration in creating our strategy. The system architecture presented in Figure 1 is organized into three layers: *i)* user interface; *ii)* servers and *iii) knowledge layer*.

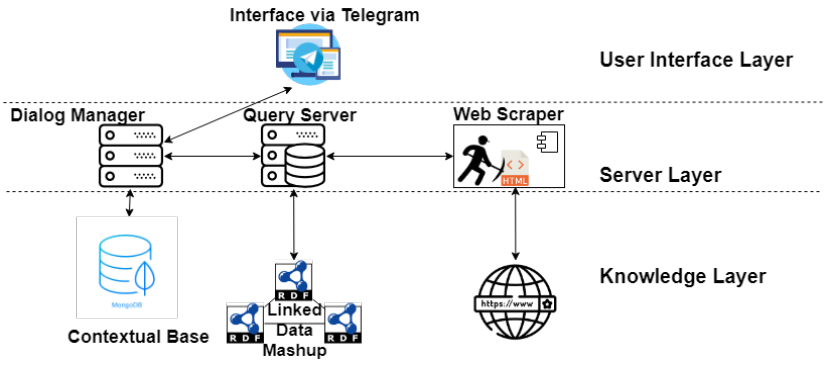

Fig. 1. System architecture

In the first layer, the **User Interface** provides a mean for user interaction through the instant messaging application *Telegram*. The reason for choosing Telegram as a channel for the system is that it already has a large user base in conjunction with another advantage. It has a straightforward API that allows the creation and use of *chatbots*.

The second proposed layer is the **Server Layer**. This layer is responsible for processing requests and responding, being composed of three main components:

—**Dialog Manager:** The *Dialog Manager* (DM) is the module responsible for receiving, processing, and interpreting user requests, controlling the flow of the dialog. This component is responsible for classifying the user intent and deciding the next action to be taken by *chatbot*. This module connects directly to the contextual base and query server. The first acts as the chatbot memory, allowing it to take into account past interactions for making decision. The second act as access to knowledge of *chatbot*, allowing it to search the requested information;

—**Query Server:** The Query Server (QS) is responsible for receiving a request for information sent by DM, being able to consult the KB directly or forward the request to the Web scrapper. In the first case, QS creates SPARQL queries that will be performed on the KB LDM *endpoint*;

—**Web Scrapper:** It is comprised of *Web scrappers* responsible for accessing Web pages and mining the information requested by QS. This process is done at runtime, which ensures constant updating of the returned information.

The third and last layer is **Knowledge Layer**. This layer is responsible for storing the knowledge required for query interaction and resolution. This layer is also made up of three components:

—**Contextual Base:** Responsible for storing context-relevant information, such as personal information about the user such as their first name, last name, language, and other information regarding the dialog flow, keeping the current state of the conversation. Chat state is used to interpret the meaning of the message sent by the user, *e.g.*, when the user responds the requested drug name

made by *chatbot*, the system must recognize that the user message is the answer to the question previously asked so that the normal flow of dialog can continue.

—***Linked Data Mashup:*** The system uses a Linked Data Mashup (LDM) as its principal source of knowledge. LDM provides a semantically integrated view of different sources, making them available through a single access method (SPARQL queries), represented by a single vocabulary (using an application ontology), with the sources stored as a single and in the same format (in a triplestore RDF) and merging information about the same object distributed across different sources (through merging entities). The integrated sources were the data of medicines presentation in the bases of the *Agência Nacional de Vigilância Sanitária (ANVISA)* (National Health Surveillance Agency, in English) and the source RDF, *SIDER*, from the project BIO2RDF [Belleau et al. 2008];

—**Drug pricing sites:** Although the LDM already have drug pricing information, it only refers to the maximum prices allowed by law, but not the actual prices. In this way, prices applied by resellers are obtained directly from the information published on the *Web* using Web scrappers that retrieve prices in real-time. That ensure that offers are updated and valid. We currently use the portal *Consulta Remédios*[5]. The *Consulta Remédios* is a portal that aggregates information about drugs and its prices provided by various pharmaceutical chains and is updated in real-time with drug offerings.

5. LINKED DATA MASHUP CONSTRUCTION

In this section, we describe the process of construction and publication of Linked Data Mashup (LDM). The integration process was based on the Linked Data Integration Framework (LDIF). LDIF suggests the following execution flow: *i)* Extraction of data sources; *ii)* Transformation of data (Triplification) and construction of exported of views; *iii)* Resolution of the identity through links *owl:sameAs*; *iv)* Data quality assessment and fusion and *v)* Data output [Schultz et al. 2012].

5.1 Selected Sources

In this article, the following criteria were used for the selection of datasets: The data should have information about drugs description, commercial drugs, drugs risks, drug's indications and finally, the data must have relevance for non-specialist users. Also, preference was given to Brazilian or Portuguese data, especially for medicines sold only in Brazil and its risks.

Based on the previously listed criteria, four different datasets were selected, three of which are available from the *Agência Brasileira de Vigilância Sanitária (ANVISA)*[6] or Brazilian Sanitary Surveillance Agency in English, and the last one belongs to the BIO2RDF project [Belleau et al. 2008]. From ANVISA, we selected:

—Consumer Drug Prices (CDP)
—Government Drug Prices (GDP)
—Drug's risks in pregnant and breastfeeding (RPB)

CPD and GPD are found in the *XLS* and *PDF* file formats in [ANVISA 2018] , wherein this work, the *XLS* version was used. Both datasets have information about allopathic drugs, such as drug name, producer, barcode, therapeutic class, presentation, the active ingredient, and prices. The only difference between them is because the former has maximum selling prices for the average consumer, while the latter has maximum selling prices for government agencies. The dataset RPB contains the risk categories of substances during the period of pregnancy and breastfeeding. This dataset can be found in the Web document [ANVISA 2010a] and is only available in unstructured *PDF*.

[5]https://consultaremedios.com.br/
[6]http://portal.anvisa.gov.br/

The last dataset selected was the *SIDER*[7], made available by the *BIO2RDF* project, and can be found in the *RDF* format. The dataset SIDER contains data about drugs, their indications, side effects, and different labels. However, the database only has data in English, not containing information about Brazilian drugs, making it necessary to translate it into Portuguese. This dataset has been selected because it contains information about the side effects of active principles, such information is needed to inform the risks of a drug.

5.2 Vocabulary

The Semantic Web technologies were used to standardize the access to the information, abstracting the different structures, physical formats, and vocabularies between the data sources. This approach uses ontologies as common standardized vocabulary to the source datasets. In the Linked Data paradigm, *OWL* ontologies [W3C 2012] are used, which provide a representation of knowledge in a taxonomic way through a hierarchy of classes and properties, one of the objectives of *OWL* is to structure data in a semantically understandable way by the machine allowing the inference of implicit information based on defined axioms. The *OWL* ontology provides a layer of semantic abstraction, allowing access to the integrated data in a transparent way to the user, in addition to using terms closer to their daily life, abstracting coded fields, and giving definitions about terms, giving so the possibility of a greater understanding to the lay user.

In our article, a vocabulary was developed based on the data dictionaries of the original datasets, as well as other sources of knowledge about the domain such as sites, books, and manuals, having, in particular, the booklet *"What we should know about drugs"* [ANVISA 2010b]. In developing the vocabulary, it was always sought to conceptualize verbatim each term used, in addition to providing different alternative nomenclatures following the non-ontological sources cited before. The *OWL* implementation can be found in [DataHub 2018].

5.3 Exported views

The exported view of a dataset consists of its representation using the vocabulary of the target ontology. In this article, we used the term *triplification* to refer to the process of generating RDF [8] triples that represent the original data using the defined target vocabulary.

For the *triplification* of datasets *CDP* and *GDP*, we imported the data into the relational database management system *PostgreSQL* [PostgreSQL 1996] , due to the existence of matured tools in the conversion of relational databases for *RDF*. In this work, we used the tool *D2RQ* [Bizer and Seaborne 2004] that performs the transformation of relational bases to RDF by mapping the original schema to the desired target vocabulary. The mapping language used was *R2RML* [W3C 2011].

For the *triplification* of the *RPB* dataset, manual conversion of *PDF* file to *CSV* was required on account of the file's internal structure. But finally, the same process previously described was used for its *triplification*.

Finally, since the *SIDER* source already exists in the *RDF* format, it was only necessary to use *SPARQL CONSTRUCT* to map the original dataset to the desired vocabulary. The result of this step was a set of four *RDF* graphs with the target vocabulary representing each original dataset. However, it is still necessary to find out which resources represent the same object between the different *RDF* graphs and merge them into one.

[7]https://download.bio2rdf.org/files/release/3/sider/sider.html
[8]https://www.w3.org/RDF/

5.4 Identity Resolution

This step is responsible for discovering which different resources represent the same object in the real world to connect them via *owl:sameAs* links. These resources can be found on a single source or between the different sources, where the direct comparison between sources is only possible because there is already a guarantee that all sources have uniform structure and vocabulary because of the ontology.

For this step, the tool *Silk* [Volz et al. 2009] was used. *Silk* uses user-specified rules to discover and generate *links* between resources. For this work, simple rules, such as strings treatment, comparison of values and averages, were used in general. For the most part, the defined rules have used the *dc:title* attribute that represents the title or nomenclature of the resource.

Links were generated between the resources of the Drug, Laboratory, Therapeutic Class, Substance, and Presentation classes. The other classes were not considered because their mappings themselves already guarantee the creation of resources with the same URI.

5.5 Quality Evaluation and Data Fusion

After the identity resolution step, it is already possible to know through the *owl:sameAs* links which distinct resources represent the same real-world object. However, such resources are still represented by distinct *URIs*, so there is a need to merge such resources into a single one that will encompass all the properties and relationships of the originals. However, this merge may cause some problems, such as repeated values for properties, conflicting between properties values, or incorrect values. So, it's necessary to have the **quality evaluation** of the sources, this process defines the degree of priority of each source. After the quality evaluation, the **data fusion** and **data cleaning** process will be performed, choosing which information should be kept or deleted, based on the priority of each source.

For this step, we used the tool *Sieve* [Mendes et al. 2012]. For the quality evaluation phase, the metric *ScoredPrefixList* was used, where for each source a weight was given in the following order: CDP, GDP, RPB and finally *Sider*. Such order was selected using manual analysis taking into account the quality of how the data is structured, the scope of data, number of conflicts in original data and the number of links created between resources of the same source. This last rule comes from the intuition that if a source has many different resources that represent the same object, then the source was constructed less rigorously, so it will have a lower priority. Fusion rules have been defined for cleaning only the classes Drug, Laboratory, Therapeutic Class, Substance, and Presentation, due to a possible fusion of such resources.

5.6 Publication of the Linked Data Mashup

At the end of the semantic integration process, we generated a dataset RDF containing the integrated view of the four original datasets, now following the same vocabulary and resources merged. This final dataset is called a *Linked Data Mashup*(LDM).

The resulting dataset was then hosted in the *virtuoso triplestore*[9] , which provides a *SPARQL* endpoint capable of responding to *SPARQL* queries via HTTP. This *endpoint* is accessed directly by the *Medibot* application. Moreover, the *dump* of the final dataset representing the LDM, in addition to the mapping files and the OWL implementation of the ontology can be accessed publicly via datahub[10].

[9]https://virtuoso.openlinksw.com/
[10]https://datahub.io/linkeddatamashupeducacional/data-med/v/2

6. MEDIBOT

Although the ontology provides a layer of semantic abstraction with terms closer to the user, there are still problems in its access. To have access to the data, before it is necessary to know about the ontology's schema and knowledge about Linked Data technologies, such as *RDF*, *OWL*, and *SPARQL*. A SPARQL query can be overly complicated, requiring technical expertise on the part of the user which would go against the purpose of this work which is to universalize knowledge about drugs for users of different profiles. Therefore, in this work, a data access interface was developed via natural language through a *chatbot*, called *MediBot*[11].

MediBot is a *chatbot* for the instant messenger *Telegram*, so that it can be used both via mobile application and via the Web interface on the PC. *MediBot* was implemented in JavaScript using NodeJS. Currently, *MediBot* is able to answer questions in Portuguese. *MediBot* can is contacted via Telegram by id **@LDM_MediBot**.

MediBot has two modes of operation, the *Quick Response*, and the *Interactive Mode* modes. In quick answer mode, the chatbot seeks to match the question asked in natural language to one of the query templates defined a *priori*. The templates used in the quick response mode are defined using regular expressions. Quick response mode only supports single-stage interactions. In this mode, always there are only a single question and a single answer, where previous interactions are not considered in current computing.

The second mode, the interactive mode, is performed in a conversational style. In this mode, there is a sequence of related questions and answers, where previous interactions are considered while computing the current interaction. This mode has three distinct tasks: *Browser*, *Query* and *Price Comparison*.

6.1 Quick Response Mode

In quick response mode, *MediBot* has a set of *SPARQL* queries predefined and uses a simple regular expression evaluation approach to mapping the entry in one those predefined queries. During the input evaluation process, key terms and filtering parameters are retrieved. Key terms help classify into which type of query the entry should be mapped to, while filtering parameters are used in FILTER clauses to restrict the query result to the specific intent of the user. Finally, the *SPARQL* query is built and performed via *HTTP* on the *Virtuoso endpoint*. Moreover, the building answer process also uses answers patters predefined. Eight types of queries have been defined, which are shown in Table I. While Figure 2 presents an image of *MediBot* responding to query ‘‘*What are the risks of reopro?*” in Portuguese through the *Telegram*.

Table I. Types of queries answered by *MediBot* in quick response mode.

Type of query	**Example**
Drugs with a principle active	What are the drugs with the substance dipyrone?
Definition of terms in domain	Define therapeutic class
Informations about certain drug	Talk about the drug aspirin
Indicate the risks of a drug	What are the risks of the drug reopro?
List of drugs's presentation	What are the presentations of the drug reopro?
Information about barcode presentation	Give information about bar code presentation 7896382701801
Price of a presentation with ICMS tax in one State	What are the price with ICMS tax of presentation 7896382701801 in the state of Ceará?
The maximum prices by law of a drug	What are the maximum prices for Buscopan?

[11] https://github.com/andersonbr/websemanticabot

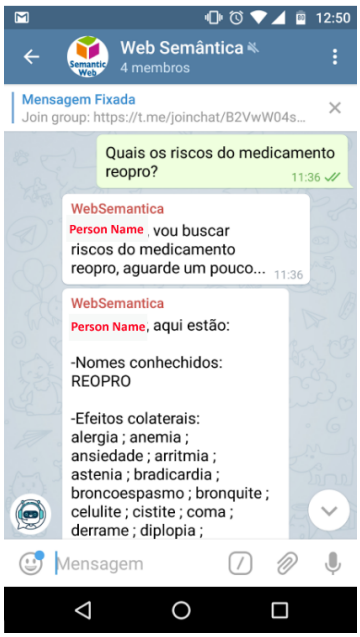

Fig. 2. Example of Query in Portuguese on Telegram about the Drug's risks.

6.2 Interactive Mode

In addition to the quick response mode, *MediBot* has an interactive mode. While the former provides quick and easy access to information, the latter offers a versatile form of access to the knowledge contained in the sources. The main difference between the two modes lies in the interactive and conversational character of the interactive mode, while the interaction of the quick responses mode is summed up to individual questions and answers. The interactive mode performs information retrieval tasks in a conversational way where the context of the conversation and previous interactions influences the results of future interactions.

The interactive mode is oriented to finite tasks, where the user starts one task at a time through the sending of messages containing pre-established keywords. Moreover, a task remains active as long as the user does not explicitly intend to finish it. However, during any point of an interactive mode task, the user can perform a quick response mode query. Meanwhile, the already started task remains in standby to be resumed any time the user wishes.

During interactive mode, three types of tasks can be performed: browser, query, and price comparison, which will be described later. However, the flow followed during the conversation is not free, following a well-defined standard flow, where chatbot and user alternate their questions and answers. For decision of which next steps to be followed during a specific task's conversation, the *MediBot* uses information about past interactions during the same task (in this case called **context**), the current point of the task (in this case called **state**) and, in cases where the chatbot expects a response from the user, the message received message (in this case called **input**).

To enable continuous interaction between chatbot and user the interactive mode was implemented following a variation of the pushdown automaton approach, where the current state of the conversation or simply state is represented as a state of the automaton, the context of the conversation is represented as the auxiliary memory of the automaton, and the input of the user is represented by the input signal of the automaton. A remarkable aspect of the implementation of *MediBot* is the fact of the possibility of state change without an explicit input sequence, this is due to the fact that previous user responses

may already have the information needed for the state change without that there is a need for the user to reissue his intention.

6.2.1 *Browser and Query Tasks.* Due to simplicity, the formal definition of the *MediBot* automaton will not be presented. However, in the figure 3 a flowchart is presented representing a summed-up view of the tasks Browser and Query performed in the interactive mode.

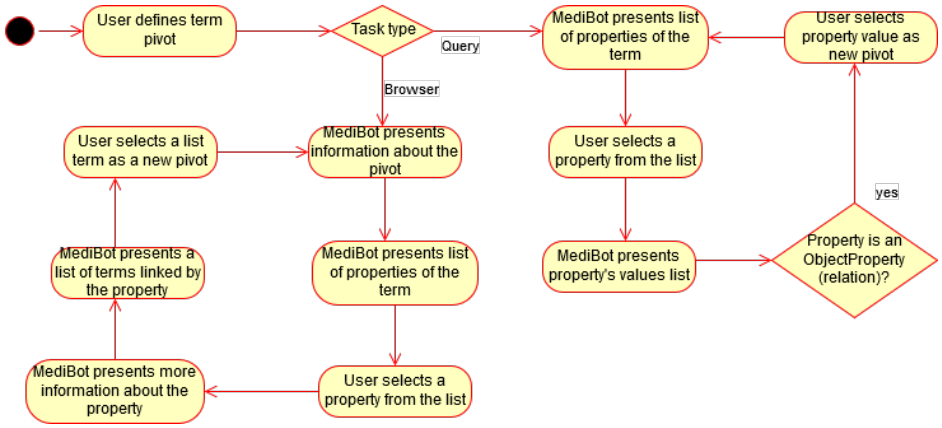

Fig. 3. Interactive mode flowchart

—**Browser Task:** Acts as an electronic dictionary on terms present in application ontology, allowing the user to navigate on its concepts. This task allows the user to navigate recursively on the existing terms in the knowledge base, including the ontology schema and its instances. When the user starts the task of browsing over a term, *MediBot* presents the different names, types and, definitions of the term, besides presenting its properties and allow the user to select one of this to be the new pivot of the task, being able to navigate on the concepts of the sources. In figure 4 is presented as an example of a browser task's interaction on Telegram.

—**Query Task:** Allows user to query property values of instances contained in LDM. This task allows the user to view data about instances contained in the knowledge base. Likewise, the browser task, the query task is also recursively interactive. During this task, the user can ask to query the properties of a given instance. Upon receiving a query from a user, *MediBot* returns a list of instance's properties, giving the user the option to select one of these to view its values. If the selected property is a relation (*owl:ObjectProperty*) *MediBot* displays the list of instances as values for the property selected, which can be selected as the new pivot for the query task. If the selected property is a simple attribute (*owl:DatatypeProperty*), it simply displays a list with the constant values for the property. In figure 5 is presented as an example of a query task's interaction on Telegram.

It is interesting to note that some steps of the query and browsers tasks are similar, such as terms list's display, these steps were implemented as a single state in the automaton, with the context being the only point of differentiation them. Another important implementation element lies in context's persistence since for being an online application and multi-user isn't feasible to keep the context in memory, so each time the context is changed or necessary it has resorted to the persistence module, MongoDB.

6.2.2 *Price Comparison Task.* In this task the *chatbot* searches the Web for the different prices offered for the same medicine in different retailers. In addition, the system also looks for the prices of

drugs that can replace the original, showing the one with the lowest prices as a possible alternative. The workflow of the price comparison activity follows these steps:

(1) The chatbot prompts the user for the drug name;
(2) User enters the full or partial name of the drug;
(3) The system searches in LDM medicines that contain the name requested by the patient in their names and presents the list of possibilities to the user;
(4) The user selects a drug M of interest from the list;
(5) The system searches the LDM for similar drugs $S_1, ..., S_n$ to M. Similar medicines are those that have the same active ingredients and effects;
(6) The system search in the *Web* price offers for drugs $M, S_1, ..., S_n$. For each M result found, it is considered a A_kM presentation, where $1 \leq k \leq RM$, with RM being the number of results found for the search using M as the keyword. Also, for every S_j result found, it is considered a A_kS_j presentation, where $1 \leq k \leq RS_j$, where RS_j is the number of results found for the search using S_j as the keyword and $1 \leq j \leq n$.
For each result found $(A_1M, ..., A_{RM}M, A_1S_1, ..., A_{RS_1}S_1, ..., A_1S_n, ..., A_{RS_n}S_n)$ the system extracts information about drug name, description of presentation, lowest price, name of lowest priced retailer, lowest price dealer offer website, drug description and how to use it;
(7) With the extracted drug information, the system groups the drugs by presentations. For each G_{A_kM} is composed of one representative, this is A_kM, and presentations of similar medications. $A_kS_j \in G_{A_kM}$, in other words, a presentation A_kS_j, belongs to the presentation group A_kM, when the *levenshtein* distance between the description of the presentation of A_kS_j and of the presentation of A_kM is less than or equal to a *threshold* t:
$levenshtein(presentation(A_kS_j), presentation(A_kM)) \leq t$;
(8) For each group G_{A_kM} the system chooses a better alternative S_kM, where $\{S_k | A_kS_j \in G_{A_kM}$ & $price(A_kS_j) = priceMin(G_{A_kM})\}$, in other words, S_k is the presentation of a similar medicine with the lowest price within G_{A_kM};
(9) Finally, the *chatbot* shows a list of presentations $(G_{A_1M}, ..., G_{A_{RM}M})$ sorted in ascending order by the price of each group's representative A_kM and showing S_k when $price(S_k) < price(A_kM)$.

Algorithm 1 presents the algorithm executed in steps (7) and (8) of the price comparison workflow. The logic behind the algorithm is as follows: The algorithm receives as input the set of items found using the original drug name as the (original) search term, the set of items found using each of the similar drug names as the search term (similars), tolerance *threshold* for drug names (t_name) and tolerance *threshold* for drug presentation descriptions (t_presen). Then, in lines 2 and 3 the algorithm filters out possible noises from the original drugs and groups them by presentations. Subsequently, for each possible similar drug (filtered noise in 6) the algorithm classifies it into one of the original groups, based on the distance (currently *levenshtein*) from the drug presentation to the presentation of the representative of each group (lines 9 - 14). When a drug is added to a group (line 14) it is decided if it is the similar representative or not, this happens when the relevance value of the item being evaluated is higher than that of the current representative. The relevance of an item is calculated as: $R(I) = 1/(WP*max(I.price, 0.001)) + 1/(WD*max(I.dist, 1))$, where $WP+WD = 1$. This measure attempts to weigh the importance of the lowest price and the shortest distance for the relevance of an item in the group.

Algorithm 1. Item grouping and price comparison algorithm.

```
input: original,similars,t_name,t_presen
itens_original = filter(original.itens,original.name,t_name)
groups = Group_By_Presentation(itens,t_presen)
for similarGroup in similars:
    for similar in similarGroup.itens:
        if(distance(similar.name,similarGroup) <= t_name):
            best_group = null
            best_dist = infinity
            for group in groups:
                if(distance_group(group,similar,t_presen) < best_dist):
                    best_group = group
                    best_dist = distance_group(group,similar)
            if(best_group != null):
                best_group.add(similar,best_dist)
return groups
```

An example of a price comparison query made by *MediBot* can be seen in Figure 6.

7. EVALUATION

In this article, we performed two evaluations to measure *MediBot*'s usability and effectiveness. The first, usability evaluation was done with volunteers, where they were asked to use the tool and report their experience of use. The second sought to measure the effectiveness of chatbot in the task of comparing drug prices. The effectiveness assessment was performed manually by the project developers.

7.1 Usability Evaluation

In this article, we selected the task-based evaluation method described in [Konstantinova and Orasan 2013] to measure the *MediBot*'s usability degree. We use the following metrics proposed by [Da Costa et al. 2019]:

—**UH2 - CORRESPONDENCE BETWEEN THE APPLICATION AND THE REAL WORLD:** According to UH2 - CORRESPONDENCE BETWEEN THE APPLICATION AND THE REAL WORLD, the application must speak the users' language and not in the technical terms of the system. The application must follow the real world's conventions and display the information in a logical and natural order. We built some interaction flows that reproduce a conversation in the real world. The purpose of this is to make the flow of dialogue more natural through real-world elements and recognizable concepts. The main advantage of this approach is that when recognizing the concepts of the real world in the application, the user will have a lower barrier to adapt to the use of the system and correctly interpret the information provided by the system as they will be presented in a logical and natural order.

—**UH3 - USER CONTROL AND FREEDOM:** According to UH3 - USER CONTROL AND FREEDOM, the application must allow the user to undo and redo their actions for precise navigation. It must provide the user with an option to get out of undesirable states of the system. Our interaction flow always allows the user to end a conversation and or return to the previous flow. This strategy aims to enable the user to move from an unwanted state to the desired state quickly. Besides, the application must allow the user to undo and redo their actions and intuitively.

—**UH8 - EFFICIENCY OF USE AND PERFORMANCE:** According to UH8 - EFFICIENCY OF USE AND PERFORMANCE, our system uses the Telegram platform to upload and present the requested content. The tasks and transitions are made available smoothly and without interruptions,

Table II. Evaluation result.

ID_Question	Question	Mean Time(s)	Success Rate(%)	Difficult
Q001	What is a black box remedy?	30,53	100	1
Q002	What are the risks of Tylenol?	51,2	100	1
Q003	Which drugs have the same active ingredient as Buscopan?	142,10	80	2
Q004	What is Buscopan's maximum price?	358,96	20	2
Q005	What is the relationship between a substance and a presentation?	152,05	70	3
Q006	Which state has the lowest maximum price for the prescription drug orencia?	320,34	10	3
Mean		175,86	63,33	

according to the message exchange pattern established by the social network. Tasks are short, with limited options for execution

In this method are defined sets of tasks that are then requested to be performed by volunteers.

To evaluate the practical usefulness of *MediBot*, a set of 6 questions about the information contained in the sources was elaborated. The questions were asked for not involved with the project volunteer users who had to use the tool to respond to them. The questions can be divided into three levels of difficulty: (1) Easy, can be answered with a single interaction with the chatbot; (2) Medium, that can be answered with at least two interactions; and (3) Difficult, requiring more than two interactions with the chatbot.

The questions were asked through face-to-face interviews with volunteers. Ten volunteers, four men, and six women participated in the study. The average age of the participants was 31 years, having a minimum age of 22 years and a maximum of 63 years. Also, two of the participants had technical knowledge in information technology, among them one knowing about ontologies.

As evaluation criteria, it was using three aspects. The first criteria were the time needed to solve the questions, the second was the rate of correctness, and finally, the personal opinion of each evaluated that entered as a subjective criterion. In table II the result of the evaluation for each question and the final average is presented.

In the criterion of success rate dropouts were counted as an error; however, the time of such cases did not enter the meantime because it would cause distortions.

It is noteworthy that only queries Q001 and Q002 were able to be performed without prior knowledge about the ontology and the types of queries and their flows, while the others needed queries on such information. This fact was already expected, since *MediBot* has a limited set of pre-defined questions (which includes queries Q001 and Q002), whereas query and browser tasks require a correct starting point to be useful.

Another interesting point is that queries Q004 and Q006 took considerably longer, in addition to having a lower success rate. This fact can be explained because their answers are not represented in a factual way at the base. In the case of Q004, there was a need for a comparison operation, which in general users tried to compare all current prices, which required many interactions, leading to a high dropout rate. There was the possibility to remove the number of options taking in characteristics of the presentation, such as quantity, route of administration, and others. Already in the case of query Q006, there was a need to have an understanding of how the state was related to price, where once again users attempted to make all comparisons. However, such a question could be resolved only by

looking at the lowest value for taxes and which states adopted it. During the reporting of opinions about the use of the tool, the main points presented were that ontology's image and examples of queries were handy - besides, the preference for quick questions about query and browser tasks.

7.2 Effectiveness Assessment of Price Comparison Mode

Table III. Results of preliminar evaluation.

Drug	Found	Minimum Price	Similar Found	Similar Price	Price Difference	Maximum Price
ROSUCOR	1	32,65	1	23,46	9,19	48,51
SORINE SSC	1	15,37	0	-	-	35,33
RELAXMED	1	3,99	1	1,9	2,09	5,96
VICK VAPORUB	1	8,04	0	-	-	23,86
DROPROZINA	1	6,35	1	6,34	0,01	6,35
CORISTINA D	1	6,19	0	-	-	9,09
BENEGRIP	1	6,6	0	-	-	9,3
BUSCOPAN	1	10,91	1	7,49	3,42	14,4
MERTHIOLATE	1	13,28	1	5,07	8,21	17,92
RENALAPRIL	1	8,67	1	1,8	6,87	17,31
DORFLEX	1	3,99	0	-	-	15,02
XARELTO	1	66,9	0	-	-	66,9
SELOZOK	1	20,32	0	-	-	57,68
NEOSALDINA	1	1,52	0	-	-	29
TORSILAX	1	1,9	0	-	-	36,52
ARADOIS	1	43,84	1	28,88	14,96	135,56
GLIFAGE	1	15,69	1	4,69	11	29,5
ADDERA D3	1	18,4	1	15,46	2,94	114,04
ANTHELIOS	0	-	-	-	-	-
BUSCOPAN COMPOSTO	1	10,91	0	-	-	14,98
TOTAL AVERAGES	0,95	15,55368421	0,474	10,56555556	6,521111111	36,17
TOP AVERAGES	0,9	20,38555556	0,333	14,89888889	9,633333333	55,46666667
RANDOM AVERAGES	1	11,205	0,6	7,676666667	4,965	18,803

Preliminary evaluations were performed to evaluate the effectiveness of the system in finding the lowest prices offered for a drug presentation and the lowest prices for possible alternatives. To this end, we performed the manual querying process of 20 drugs, where the first 10 (RUSUCOR-RENALAPRIL) were randomly chosen from the options catalog and the last 10 were chosen following the top of the best-selling medicines[12]. The site *Consulta Remédios* was used as a source for comparisons. In our tests we used the values of $t\_name = t\_presen = 8$ e $WP = 0.9, WD = 0.1$, giving a greater relevance to the price. These values were not explored to exhaustion, being chosen from a few tested values, thus being a point for future improvements.

In our tests (with results in Table III) we found that the price of a drug can vary significantly between different retailers, where the average for the minimum price is R$ 15.50, while the average for maximum price offers to reach R$ 36.17. In addition, in our tests, we found that the system is useful in finding the drugs you have sought, having found 90% of the best-selling drugs (TOP Averages). For similar medicines, we found an alternative in about 33% of the best-selling drugs and 60% in random drugs, which indicates that we still have plenty of room for improvement in this area. Again, regarding similar drugs, in our tests, we observed that even if the system did not make the recommendation, it would be possible. However, it didn't happen because the presentation description presents the same words, but in different orders, which was not captured by the metric purely syntactic grouping, thus showing this a point for future improvements.

[12]https://guiadafarmacia.com.br/interfarma-faz-lista-dos-dez-medicamentos-mais-vendidos-no-brasil/

8. CONCLUSION AND FUTURE WORK

In this article, we presented the *MediBot*. *MediBot* is a chatbot for querying drug information. The system presented aims to act as a single access point for retrieving information, its prices, and risks, about drugs in a natural and simplified way. Therefore, democratizing access to knowledge and promoting conscious consumption of drugs.

The system uses as a source of knowledge a Linked Data Mashup (LDM) and information extracted from the Web in real-time. The LDM semantically integrates heterogeneous *datasets* from different sources published on the Web, allowing a homogeneous and integrated view of the data. Moreover, the system recovers the prices offered for medicines in the main pharmaceutical chains of the country.

In this article, we focus on presenting the system architecture, the construction process of the Linked Data Mashup and MediBot's activity modes, with focus on the new task of Price Comparison. The price comparison task was implemented following the demands of the volunteers in the chatbot usability process. This task presents the best prices for medicines and their best possible substitutes extracted in real-time from Web. In our preliminary tests, we proved the system's effectiveness in finding the lowest prices for a drug submission, achieving a success rate of 90% for the best-selling drugs in the country. However, the rate of substitute recommendations was only around 33% for the same drugs, thus indicating the way for possible improvements. In the drug presentation grouping mechanism that considers only syntactic aspects of the drug description presentation, being sensitive to variations in its writing.

Finally, as future work, we must implement usability tests with different user profiles. In addition, to improve the use of information obtained from the chatbot by specialists of different nationalities, a translation mechanism must be implemented.

9. ACKNOWLEDGEMENTS

This work was partially carried out at the Autonomous Research Center TechLab of the Autonomous University of Lisbon, Portugal, which the authors would like to thank for their support and funding.

10. APPENDIX

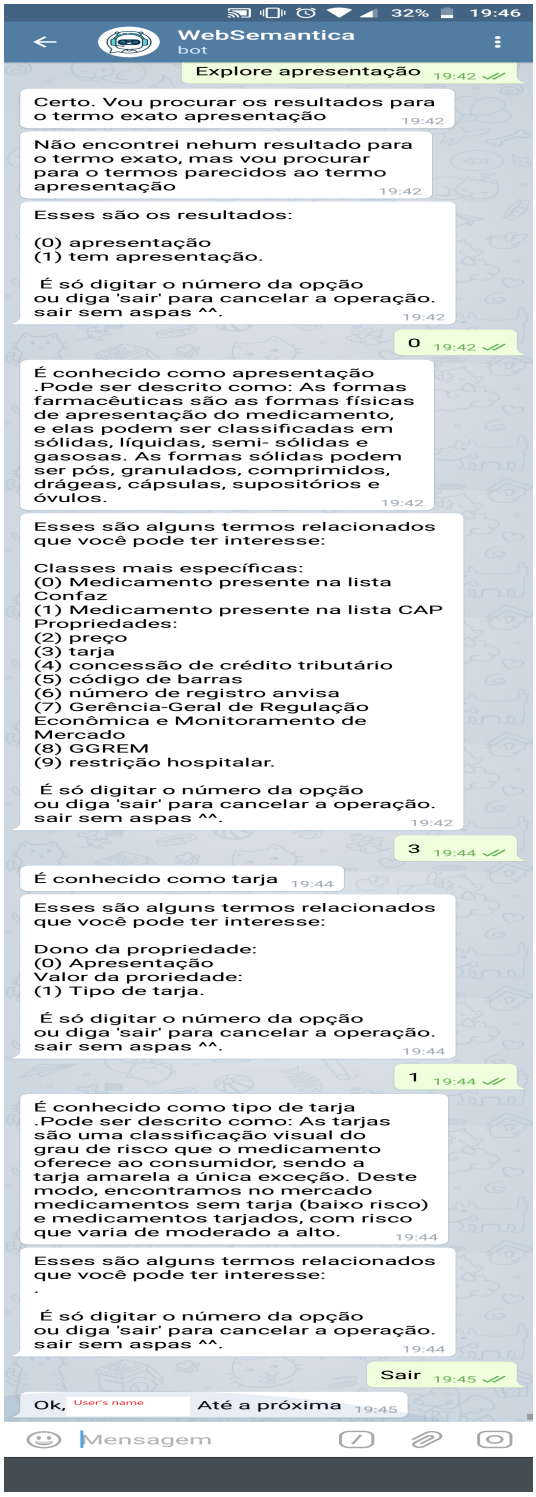

Fig. 4. Example browser task in telegram

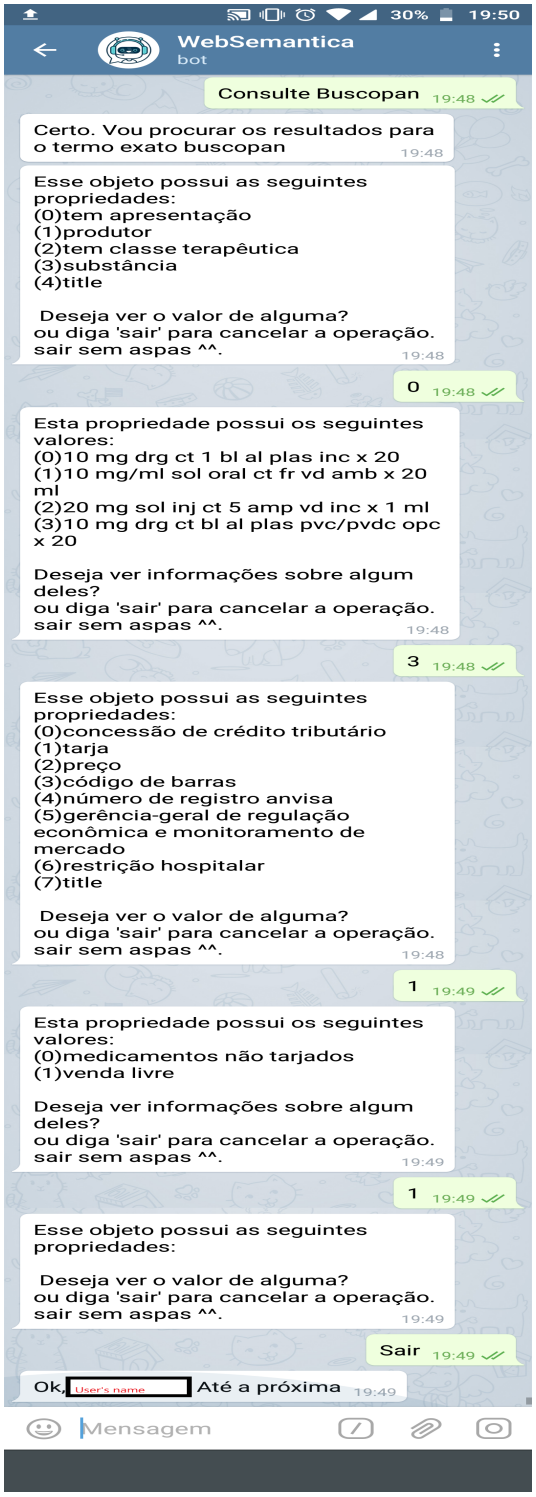

Fig. 5. Example query task in telegram

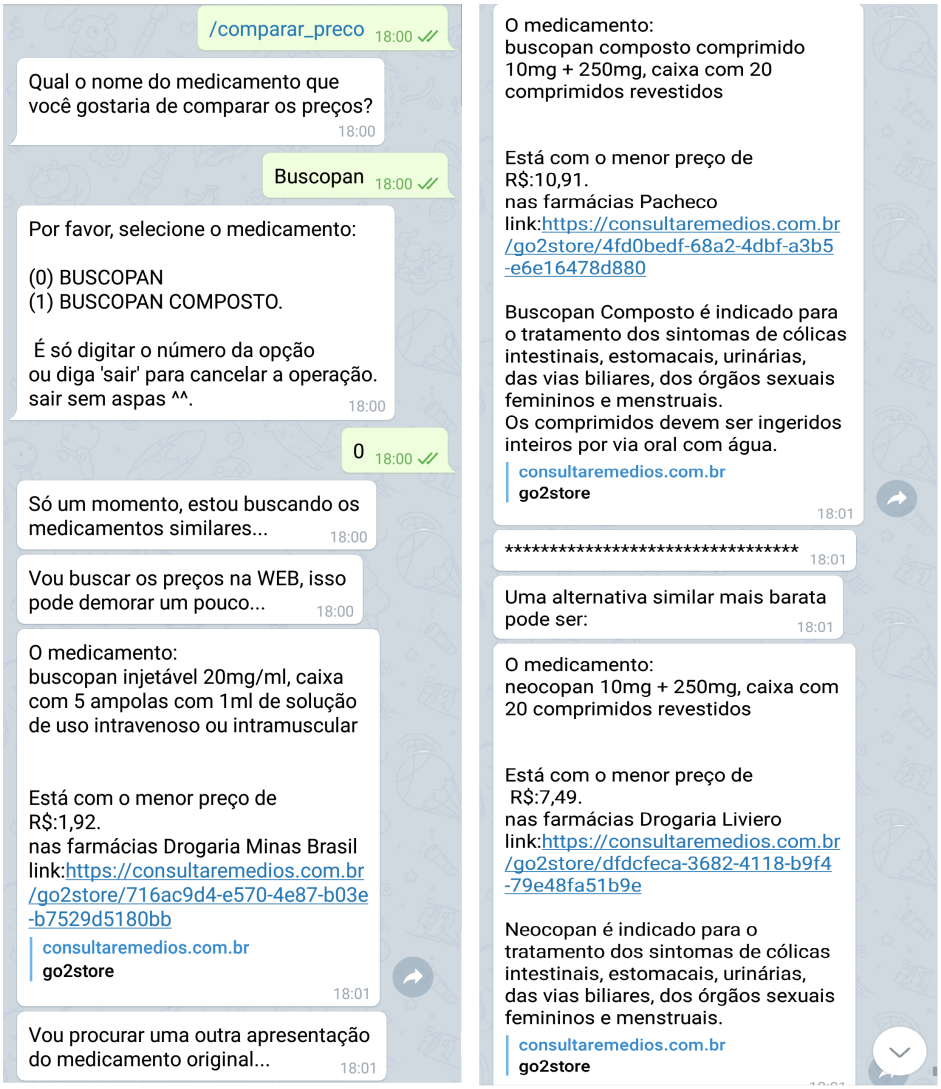

Fig. 6. Price Comparison Demostration

REFERENCES

ANVISA. gravidezconsulta pública nº 116, de 23 de dezembro de 2010. `http://portal.anvisa.gov.br/documents/33880/2561889\\/116.pdf/2292b730-2bd5-4acc-b378-10682b1fc344?version=1.0`, 2010a. Accessed: 2018-09-17.

ANVISA. O que devemos saber sobre medicamentos. `http://www.vigilanciasanitaria.sc.gov.br/index.php/\\download/category/112-medicamentos?download=102:cartilha-o-que-devemos-saber-sobre-medicamentos-anvisa`, 2010b. Accessed: 2018-09-17.

ANVISA. Preços medicamentoslistas de preços de medicamentos. `http://portal.anvisa.gov.br/listas-de-precos`, 2018. Accessed: 2018-09-17.

Aquino, D. S. Por que o uso racional de medicamentos deve ser uma prioridade?. *Ciência & Saúde Coletiva* vol. 13, pp. 733–736, 2008.

Avila, C., Calixto, A., Rolim, T., Franco, W., Venceslau, A., Vidal, V., P. V., and Moura, F. Medibot: An ontology based chatbot for portuguese speakers drug's users. In *Proceedings of the 21st International Conference on Enterprise Information Systems - Volume 1: ICEIS,*. INSTICC, SciTePress, pp. 25–36, 2019.

Avila, C. V. S., Rolim, T. V., da Silva, J. W. F., and Vidal, V. M. P. Medibot: Um chatbot para consulta de riscos e informações sobre medicamentos. In *Anais Estendidos do XIX Simpósio Brasileiro de Computação Aplicada à Saúde*. SBC, pp. 1–6, 2019.

Belleau, F., Nolin, M.-A., Tourigny, N., Rigault, P., and Morissette, J. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics* 41 (5): 706–716, 2008.

Berners-Lee, T., Hendler, J., and Lassila, O. The semantic web. *Scientific american* 284 (5): 34–43, 2001.

Bizer, C., Heath, T., and Berners-Lee, T. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*. IGI Global, pp. 205–227, 2011.

Bizer, C. and Seaborne, A. D2rq-treating non-rdf databases as virtual rdf graphs. In *Proceedings of the 3rd international semantic web conference (ISWC2004)*. Vol. 2004. Proceedings of ISWC2004, 2004.

Chowdhury, G. G. Natural language processing. annual review of information science and technology, 2003.

Da Costa, R. P., Canedo, E. D., De Sousa, R. T., Albuquerque, R. D. O., and Villalba, L. J. G. Set of usability heuristics for quality assessment of mobile applications on smartphones. *IEEE Access* vol. 7, pp. 116145–116161, 2019.

Dale, R. The return of the chatbots, 2016.

DataHub. Datahub datahub medicamentos. `https://datahub.io/linkeddatamashupeducacional/data-med/v/2`, 2018. Accessed: 2018-09-18.

G1. Remédios têm diferença de preços que pode passar de 5.000%. `http://glo.bo/29qiFAg`, 2016. Accessed: 2019-07-18.

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., and Slocum, J. Developing a natural language interface to complex data, 1978.

Hoang, H. H., Cung, T. N.-P., Truong, D. K., Hwang, D., and Jung, J. J. Retracted: Semantic information integration with linked data mashups approaches. *International Journal of Distributed Sensor Networks* 10 (4): 813875, 2014.

Jovanovik. Consolidating drug data on a global scale using linked data. *Journal of Biomedical Semantics* 8 (1): 3, 2017.

Kazi, H., Chowdhry, B., and Memon, Z. Medchatbot: An umls based chatbot for medical students. *International Journal of Computer Applications* 55 (17), 2012.

Konstantinova, N. and Orasan, C. Interactive question answering. In *Emerging Applications of Natural Language Processing: Concepts and New Research*. IGI Global, pp. 149–169, 2013.

Mendes, P. N., Muhleisen, H., and Bizer, C. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*. ACM, pp. 116–123, 2012.

Natsiavas, P., Maglaveras, N., and Koutkias, V. Evaluation of linked, open data sources for mining adverse drug reaction signals. In *International Conference on Internet Science*. Springer, pp. 310–328, 2017.

Neto, L. E. T., Vidal, V. M. P., Casanova, M. A., and Monteiro, J. M. R2rml by assertion: A semi-automatic tool for generating customised r2rml mappings. In *Extended Semantic Web Conference*. Springer, pp. 248–252, 2013.

Nováček, V., Vandenbussche, P.-Y., and Muñoz, E. Using drug similarities for discovery of possible adverse reactions. In *AMIA Annual Symposium Proceedings*. AMIA, 2017.

Petherbridge, N. Rivescript what is rivescript, 2019. Accessed: 2019-01-29.

PostgreSQL, G. D. G. PostgreSQL the world's most advanced open source relational database. `https://www.postgresql.org/`, 1996. Accessed: 2018-09-30.

Radesko, L.; Mladenic, D. A survey of chatbot systems through a loebner prize competition, 2012.

Radesko, L. and Mladenic, D. Natural language tool/dialog manager, 2018. Accessed: 2018-12-12.

Schultz, A., Matteini, A., Isele, R., Mendes, P. N., Bizer, C., and Becker, C. LDIF - A Framework for Large-Scale Linked Data Integration. In *21st WWW, Developers Track*. pp. to appear, 2012.

Schyve, P. M. Language differences as a barrier to quality and safety in health care: the joint commission perspective. *Journal of general internal medicine* 22 (2): 360–361, 2007.

Vega-Gorgojo, G. and Slaughter, L. Easy-to-use semantic search of pharmacological data. In *Proceedings of the 9th International Semantic Web Applications and Tools for the Life Sciences Conference*, 2016.

Volz, J., Bizer, C., Gaedke, M., and Kobilarov, G. Silk-a link discovery framework for the web of data. *LDOW* vol. 538, 2009.

W3C. R2rml: Rdb to rdf mapping language, 2011.

W3C. OWL web ontology language. `https://www.w3.org/OWL/`, 2012. Accessed: 2018-09-21.

WHO. The rational use of drugs: report of the conference of experts. *World Health Organization*, 1987.