A Chatbot for Searching and Exploring Open Data: Implementation and Evaluation in E-Government

IVÁN CANTADOR, Escuela Politécnica Superior Universidad Autónoma de Madrid, Spain
JESÚS VIEJO-TARDÍO, Escuela Politécnica Superior Universidad Autónoma de Madrid, Spain
MARÍA E. CORTÉS-CEDIEL, Facultad de Ciencias Políticas y Sociología Universidad Complutense de Madrid, Spain
MANUEL PEDRO RODRÍGUEZ-BOLÍVAR, Facultad de Ciencias Económicas y Empresariales Universidad de Granada, Spain

In this paper, we present a chatbot to access open government data. Differently to similar systems reported in the research literature, the developed chatbot not only allows searching for data collections, but also exploring information within the collections. The exploration is done via complex queries that are easily built by non-expert users through a natural language conversation. Moreover, as another novel, differentiating contribution, we report a conducted user study aimed to evaluate the chatbot according to the achievement of a number of public service values, as well as measuring distinct objective and subjective metrics. Experimental results show that the proposed system outperforms traditional methods followed in open data portals.

CCS Concepts: • **Applied computing → E-government**; • **Information systems → Search interfaces**; • **Human-centered computing → Natural language interfaces**; **User studies**.

Additional Key Words and Phrases: chatbots, open data, e-government, open government, public values

ACM Reference Format:
Iván Cantador, Jesús Viejo-Tardío, María E. Cortés-Cediel, and Manuel Pedro Rodríguez-Bolívar. 2021. A Chatbot for Searching and Exploring Open Data: Implementation and Evaluation in E-Government. In *dg.o '21: 22nd Annual International Conference on Digital Government Research, June 9–11, 2021, Seoul, Omaha, Nebraska, USA.* ACM, New York, NY, USA, 18 pages. https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

1 INTRODUCTION

Open government seeks to strength democracy through a transparent, collaborative and participatory government [13]. A transparent government encourages and promotes accountability to the public, and provides information on what it is doing and on its action plans. This public information –in the form of documents, proceedings, data and processes– should be accessible in a simple and clear way, enabling citizens to conduct a control and scrutiny of government activities, as well as creating economic and social value. A collaborative government involves and commits a variety of stakeholders in the Public Administration's work. It allows cooperation not only with citizens, but also with companies, associations and other actors, and promotes joint work between public employees. This collaborative environment favors the right of citizens to actively participate in shaping public policies, allowing the Public Administration to

Manuscript submitted to ACM

benefit from the citizens' knowledge and experience. It thus encourages actions that increase the prominence and involvement of citizens in public affairs, and engages political actors with their fellow citizens [6].

An emergent development in open government relates with the use of information systems that enable citizens to become more directly involved in public governance. Some refer to this phenomenon as *e-participation*, which has been defined as "the use of information and communication technologies to broaden and deepen political participation by enabling citizens to connect with one another and with their elected representatives" [15]. In this context, technological solutions have favored the creation and implementation of new processes and public services, entailing significant improvements on efficiency, effectiveness and quality, as well as new forms of transparency, collaboration and participation.

Among the existing e-participatory approaches, Open Government Data (OGD) represents one of the most widely adopted worldwide [9]. According to the Open Definition,[1] "a piece of data is open if anyone can freely access, use, modify and share it for any purpose, subject, at most, to requirements that preserve provenance and openness." As agreed in the literature [10] and stated by the OECD,[2] OGD promote transparency, accountability and value creation. On the one hand, by making their datasets publicly available, institutions become more transparent and accountable to citizens. On the other hand, by facilitating the use, reuse and free distribution of datasets, governments foster business creation and innovative, citizen-centred digital applications and services.

In addition to *accessibility* (i.e., the data have to be freely accessed by users) and *right* (i.e., the data have to be released with under certain licenses that softly bound its usage, transformation and distribution), open data should also grant *machine readability*, i.e., they should be processed by machines. For this purpose, open data are commonly published as files with structured data on standardized formats, such as CSV (Comma Separated Values), XML (Extensible Markup Language), and RDF (Resource Description Framework), to name a few. Following this schema, international, national, regional and municipal public institutions create and maintain central web portals where they regularly publish their datasets. Examples of these portals are the European Union Open Data Portal,[3] the Open Data Portal of the Government of Spain,[4] or Madrid City Council's Open Data Portal.[5]

Open data portals are designed to enable the general public to access the open data collections. However, in general, available open data sometimes are not linked/well structured, which makes difficult to find the information [28]. The information in these portals depends on several determinants, including the department characteristics and the administrative capacity of governments [29]. In addition, open data portals are mainly built on finding data collections (files) and not on the search of information included into such collections [19]. As stated by the European Commission[6], 73% of the open data consumers characterize finding data as difficult or very difficult.

Attempting to solve these difficulties, portals provide descriptions and metadata about the datasets, as well as search functionalities that allow looking for data collections of interest. As classical web search engines, searches in open data portals are done by users through keyword-based queries, which are aimed to retrieve those collections that contain certain key terms in their titles, keywords and descriptions. Portals, in contrast, do not provide mechanisms to query, explore, visualize and analyze the data of the collections [10]. Once a collection of interest is found, a user has to

[1]Open Definition, https://opendefinition.org
[2]OECD reports on Open Government Data, https://www.oecd.org/digital/digital-government/open-government-data.htm
[3]EU Open Data Portal, https://data.europa.eu/euodp
[4]Spanish Open Government Data Portal, https://datos.gob.es
[5]Madrid City Council's Open Data Portal, https://datos.madrid.es
[6]Barriers in working with Open Data, https://www.europeandataportal.eu/en/highlights/barriers-working-open-data

download the associated file, open it with an appropriate software, and process and filter the data, which may be too time consuming and may require advanced knowledge and expertise from the user.

User-friendly applications built upon open data for easy access and exploration are thus required. Facing this need, recent work has focused on the use of conversational agents to querying data. Conversational agents are information systems that communicate with humans through natural language, in spoken or written forms [17]. Chatbots —also referred as chatterbots— can be seen as a particular type of conversational agents in which human-machine interaction is textual, rather than spoken [27].

Through a natural language dialogue, chatbots allow the users to maintain an interaction based on the ask-get response paradigm, which nowadays is widely extended in instant messaging applications, such as WhatsApp[7] and Telegram[8].

Chatbots have a number of advantages as well as challenges in comparison with other interfaces [11]. First, they are light-wight and inexpensive to implement. The recent, remarkable advances in speech recognition and natural language processing (NLP) by data-driven machine learning methods (deep neural networks, in particular), have led to an increased improvement and availability of conversational system development frameworks [7], both commercial (e.g., Google DialogFlow[9]) and open source (e.g., Rasa[10]). Second, they are intuitive and require none or little training, since they communicate with users through natural language and are deployed in well-known, widely used instant messaging applications and social networks (e.g., Twitter).

In the digital government research literature, there are a number of chatbots that have been proposed for providing support to access OGD. Some of these systems have focused on searching data sources and collections [11, 19]. However, as explained above, working with raw data remains challenging for non experts, and progress has to be done not only to ease the task of finding the right dataset and understand its structure, but also to actually work with it. Other systems address this challenge, allowing the user to ask for information within data collections [2, 23, 25]. The questions are, nonetheless, limited and domain-dependent.

Differently to previous research work, in this paper we present a chatbot to access open government data which allows the user not only to search for data collections, but also to explore the information within such collections. The exploration is done via complex, domain-independent queries for relational databases which are dynamically and easily built by a non-expert user through a natural language conversation.

Moreover, in the research literature, chatbots for OGD access have been presented with particular examples and case studies without rigorous evaluations reported. In fact, from the surveyed papers, only Neumaier et al. [19] depict a usability study. As a novel contribution of our work, in this paper, we also present an evaluation instrument based on the framework recently proposed by Makasi et al. [16], aimed to assess chatbot-mediated public services. The instrument is utilized in a user study for evaluating our chatbot, in addition to other subjective and objective measures. Our experimental results show that our chatbot outperforms traditional methods followed in open data portals.

The remainder of the paper is organized as follows. Section 2 surveys existing research work on chatbots for open government data access. Section 3 introduces main concepts about relational databases which are needed to understand our chatbot, presented in Section 4. Next, Sections 5 and 6 describe the developed evaluation instrument and user study and discuss the achieved results, respectively. Lastly, Section 7 ends with some conclusions and future research lines.

[7]WhatsApp Messenger, https://www.whatsapp.com
[8]Telegram Messenger, https://telegram.org
[9]Google DialogFlow, https://cloud.google.com/dialogflow
[10]Rasa open source conversational framework, https://rasa.com

2 RELATED WORK

In the seek of improved and new citizen services and government functions in the public sector, more sophisticated Artificial Intelligence (AI) applications have emerged. In particular, the combination of OGD and AI is crucial to generate more and better value from the data [8]. Hence, governments are developing data access systems that make use of conversational AI and are integrated in commercial voice assistants [12].

In the e-government research literature, chatbots have mainly targeted the automated attendance of citizens on a variety of public services [1, 3, 14] and to allow citizen consultation and participation [4, 26]. A few preliminary chatbots have been proposed to assist on the search and exploration of OGD. We can classify them in three groups.

A first group is composed of *information retrieval* chatbots that allow the user to search for open data collections through keyword-based queries, as done in traditional web search engines. For such purpose, they make use of an index that maps key terms to collections. Such terms are, in general, keywords extracted from the collections titles, descriptions and metadata. In this context, Neumaier et al. [19] present a cross-lingual chatbot built upon the Microsoft Azure Bot framework[11] and connected to both Facebook Messenger and Skype platforms. The chatbot has access to more than 18K collections with OGD in different languages: English, French, Spanish, Italian, Portuguese, German and Finish. The used index contains concepts from the Schema.org[12] that are mapped to BabelNet[13] entities linked to terms in the collections titles, descriptions and keywords. Via user queries associated to the Schema.org vocabulary, the chatbot retrieves collections, but is not able to search for information within the collections. Keyner et al. [11] present a chatbot implemented with the Rasa framework which follows a two-step conversational search where the user indicates the topic of interest (e.g., culture) and the location (e.g., Vienna) of the indexed open data collections to retrieve. For both cases, keyword-based queries are used. As in [19], the chatbot only retrieves the titles and links of (top-N) matching collections, and not data from the collections.

A second group is represented by conversational systems that transform the user's intents expressed in a conversation into formal queries that are launched against *relational databases* storing OGD. These databases are composed of interconnected tables associated to entities (e.g., public employees). A table has a number of columns (a.k.a. fields) that correspond to entity attributes (e.g., name, surname, birth date, and salary). The rows (a.k.a. records) of a table are instances/individuals of the associated entity having particular attribute values. In each table, there are fields called "primary keys" that represent identifiers of the records, and fields called "foreign keys" whose values are primary keys of records in the same or other tables. To retrieve information from tables, queries written in Structured Query Language[14] (SQL) are used. Among other issues, these queries allow selecting certain fields, filtering records satisfying some constraints, and sorting and grouping results according to given criteria. In a series of works [22, 23], Porreca et al. present a chatbot that accesses the OpenCantieri database published by the Italian Ministry of Infrastructure and Transport. The chatbot, built upon the IBM Watson framework[15], analyzes the user's questions to identify intents and extract named entities. Each intent corresponds to a particular SQL query template, which is instantiated with the extracted entities, forming a query that is launched against the database. The authors show that their system is able to process questions such as "How much money was invested in Abruzzo in 2015?." However, they neither explain the types of questions understood by the system nor present the structure of the SQL queries that the system handles.

[11] Microsoft Azure Bot framework, https://dev.botframework.com
[12] Schema.org vocabulary, https://schema.org
[13] BabelNet multilingual semantic network, https://babelnet.org
[14] SQL standard, https://www.iso.org/standard/63555.html
[15] IBM Watson framework, https://www.ibm.com/watson

Lastly, a third group is devoted to chatbots focused on accessing *Linked Open Data*[16] (LOD), which consist of publicly available, structured data that is interlinked with other data, forming semantic networks (a.k.a. knowledge graphs) in a global database. These networks are built with standard Semantic Web technologies (e.g., RDF(S) and OWL), and are accessible through specialized query languages, such as SPARQL [5]. The networks are composed of semantic entities and relations between entities, both identified by URIs, e.g., `dbr:Spain`[17], `dbr:Madrid`, `dbo:captial`[18] and `dbp:populationCensus`[19]. Entities and literal values are linked via <`subject`, `property`, `object`> tuples, e.g., <`dbr:Spain`, `dbo:capital`, `dbr:Madrid`>, and <`dbr:Spain`, `dbp:populationCensus`, `46423064`>. In this context, SPARQL allows accessing knowledge graphs to retrieve tuples that satisfy given filtering criteria. Hence, chatbots of this group aim to extract semantic entities and properties from user conversations in order to build SPARQL queries that retrieve tuples containing such elements. In [2], Anelli et al. present Anna, a virtual assistant built upon Google DialogFlow[20] which allows exploring the knowledge graph exposed by the Digital Library of Puglia (Italy). The graph contains information associated with digital goods of the region, and is linked to external knowledge sources, such as geographical ones. The handled conversation follows a simple flow where the user is presented with certain semantic entities and properties to choose from. With the elements selected by the user, the system fills a predefined template of SPARQL query, which is finally launched against the knowledge graph to retrieve the corresponding answers (i.e., target entities of interest). Also exploiting LOD, in [25], Ronzhin et al. introduce Loki, a chatbot aimed to allow Dutch citizens to access location-based information about their buildings and neighborhoods. The system uses the Rasa framework for handling the conversations, and employs ad hoc natural language processing to transform user questions into SPARQL queries. In this case, the user questions have more flexible structures, but still are domain-dependent. The authors do not provide details about the creation of the queries.

Comparing the three types of conversational systems, we can state that *information retrieval* chatbots are the easiest to implement because they only require a search index, which can be automatically built using well-known open-source tools, such as Apache Lucene.[21] However, since they are based on simple keyword-based searches, they are not able to understand the semantics underlying the user's questions, and thus do not deal with complex, semantically rich queries. Chatbots oriented to *relational databases*, in contrast, are able to handle more fine-grained user requests represented via SQL queries. Their main disadvantage is the need for creating such queries from questions asked by users in natural language. Nonetheless, this task can nowadays be addressed through the recent machine learning-based NLP techniques integrated in commercial and open-source chatbot development frameworks. Finally, the chatbots focused on *Linked Open Data* follow the same architecture as that used by relational database systems, and thus share strengths and weaknesses of the latter ones. However, they also require the design, implementation and reuse of consensual semantic (ontological) structures upon which the knowledge graphs are built. The majority of OGD, in contrast, are published in the form of simple CSV and spreadsheet files that correspond to relational database tables. We also note that knowledge graph tuples are commonly stored as records in database tables with three fields –subject, property and object–, so conversational systems oriented to relational databases may be adapted to LOD repositories.

Considering the benefits and disadvantages explained above for the three types of systems, we advocate for implementing a relational database chatbot. Besides, differently to the surveyed systems –which are domain-dependent,

[16] The Linked Open Data cloud, https://lod-cloud.net
[17] `dbr:` stands for http://dbpedia.org/resource/
[18] `dbo`: stands for http://dbpedia.org/ontology/
[19] `dbp:` stands for http://dbpedia.org/property/
[20] Google Dialog Flow, https://cloud.google.com/dialogflow
[21] Apache Lucene indexing and search library, https://lucene.apache.org

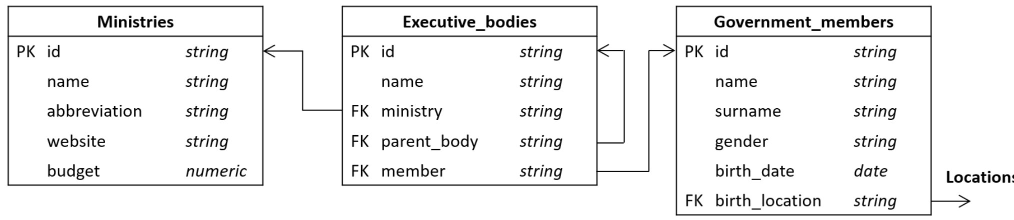

Fig. 1. Part of the schema of the example database.

handling a limited number of formal queries (either in SQL or SPARQL) with specific vocabularies (i.e., predefined entities and properties)–, our system allows the user to first search and select a collection of interest –through an index, as done by information retrieval systems–, and then maintain a conversation with the user aimed to inform about the vocabulary of the collection and dynamically build a rich SQL query to access and retrieve the relevant data from the collection.

3 RELATIONAL DATABASES AND SQL IN A NUTSHELL

Next, we provide a very brief introduction to relational databases and SQL, explaining concepts that are needed to understand the developed chatbot, presented in Section 4.

A relational database is a collection of data items with relationships between them. These items are organized in tables, with each table consisting of a set of rows and columns. The tables are associated to the *entities* (i.e, people, objects, concepts, etc.) whose information is stored in the database. Hence, each column of a table corresponds to a particular field (attribute) of its entity, and has associated certain data type (e.g., a number, a character string, or a Boolean value). Figure 1 shows part of the relational schema of a database with information about three entities: ministries, and their executive bodies and government members. Each entity is represented in a box with its fields names and data types.

A row (record) of a table, on the other hand, represents the compilation of the field values related to a particular data item of such entity. Figure 2 shows some records of the tables existing in the example database given above, instantiated with real data about the government of Spain. The table on the top gathers information (name, abbreviation, website and budget) of current ministries, the table in the middle gathers information about the executive bodies of the ministries, and the table on the bottom gathers information about all the government members.

For a given table, one or more columns can be marked as the primary key (PK) of the corresponding entity. A value of this key represents a unique identifier that each record of the table should have. Columns of the table can also be marked as foreign keys (FK) meaning that they link to values of primary keys in the same or other tables of the database, representing a mechanism to relate items of different entities. In Figure 1, each entity has an *id* field that is marked as PK, and other fields that are marked as FK, such as the *ministry* field in the *Executive_bodies* entity, which links to the *id* of the *Ministries* entity, meaning that the ministry of any record in the *Executive_bodies* table has to exist in the *Ministries* table. This and other FK relationships depicted in the figure can be checked in the example records given in Figure 2. For instance, Salvador Illa (government member with id GM4) is the Minister (executive body with id EB10) of Health (ministry with id M2).

The stored data can be accessed in many different ways without the need of reorganizing the tables of a database. This is done through formal query languages, among which SQL is the most widely used standard. The SQL language is

Ministries

id	name	abbreviation	website	budget
M1	Science and Innovation	MICINN	www.ciencia.gob.es	8,500
M2	Health	MISAN	www.mscbs.gob.es	3,524
M3	Economic Affairs	MINECO	www.mineco.gob.es	8,981
M4	Culture and Sport	MCD	www.culturaydeporte.gob.es	1,687
...	...	...	...	...

Executive bodies

id	Name	ministry	parent_body	member
EB1	Minister	M1	NULL	GM2
EB2	Secretary of State	M1	B1	GM31
EB3	General Secretary	M1	B2	GM57
EB10	Minister	M2	NULL	GM4
...	...	...	...	...

Government_members

id	name	surname	gender	birth_date	birth_location
GM1	Yolanda	Díaz	female	1971-05-06	L64
GM2	Pedro Francisco	Duque	male	1963-03-15	L11
GM3	Pablo	Iglesias	male	1978-10-17	L11
GM4	Salvador	Illa	male	1966-05-05	L29
...	...	...	...	...	

Fig. 2. Some records of the example database.

composed of a number of types of statements that entail operations for data definition (creating and modifying tables), data manipulation (inserting, updating and deleting data), data access control (managing data access privileges), and data query (retrieving data).

The data query operations have the following general query structure and syntax: a mandatory `SELECT` clause, which specifies the fields (columns) of the query results to be retrieved or `*` in case of all fields have to be returned; a mandatory `FROM` clause, which establishes the table(s) from which retrieving the data; and an optional `WHERE` clause, which sets filtering criteria to be satisfied by certain field values of the returned data. Hence, the query given below returns the name and surname of those government members who were born after 1970:

```
SELECT name, surname
FROM  Government_members
WHERE birth_date > '1970-12-31';
```

The result list of a query could be sorted (SORT BY clause) and limited (LIMIT clause). For example, the following query returns (all the fields of) the 10 ministries with highest budgets:

```
SELECT *
FROM  Ministries
ORDER BY budget DESC
LIMIT 10;
```

Moreover, a query can be applied to multiple tables linked by their primary and foreign keys. The following query returns the name of the government members with their executive bodies:

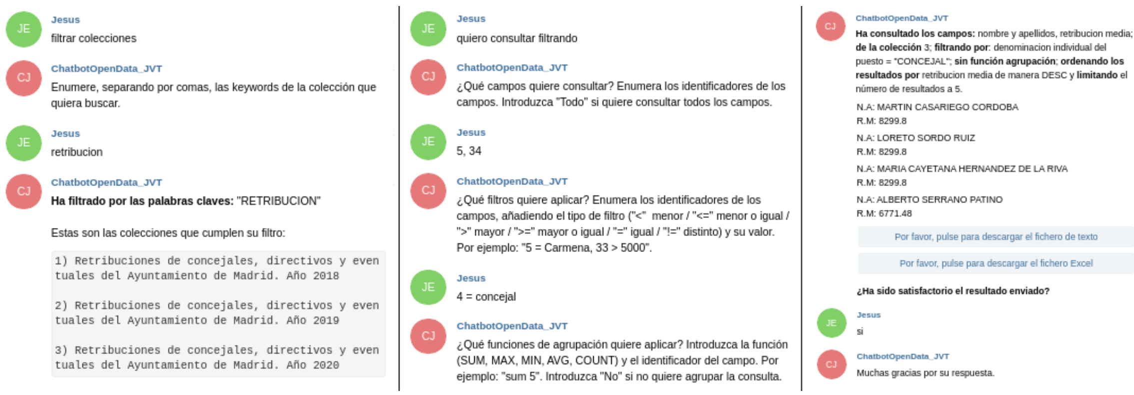

Fig. 3. Fragments of a conversation with the chatbot.

```
SELECT Government_members.name, Executive_bodies.name
FROM  Government_members, Executive_bodies
WHERE Government_members.id = Executive_bodies.member;
```

Lastly, to conclude this introduction to SQL, we comment that a query can also compute aggregation functions (e.g., COUNT, AVG, MAX, MIN) over individuals grouped by certain criteria (GROUP BY clause). For example, the query given below returns the number of ministers and their average for each gender (i.e., male and female):

```
SELECT gender,
      COUNT(*) AS number,
         AVG(DATEDIFF(YEAR, GETDATE(), birth_date)) AS age
FROM  Government_members, Executive_bodies
WHERE Government_members.id = Executive_bodies.member
      AND Executive_bodies.name = 'Minister'
GROUP BY gender;
```

4 DESCRIPTION OF THE SYSTEM

In this section, we describe the conversational flow and the architecture of the developed chatbot. Its code and data are publicly and freely available.[22] Figure 3 shows a couple of screenshots with fragments of a conversation maintained in our chatbot. We will refer to these conversation fragments next.

4.1 Conversation flow

Figure 4 shows a diagram with the conversation flow handled by the chatbot. A conversation is composed of *intents* representing different user needs (purposes or goals). An intent may be independent from the rest of intents, or should only be considered after addressing another particular intent.

In a conversation, each intent is triggered when the user produces an utterance that satisfies certain sentence pattern, which is defined in advance for the chatbot NLP model building. When an intent is triggered, the chatbot calls an external (web) service to run certain software routine. The results of the routine are returned to the chatbot, which

[22]Chabot code and data, `<URL hidden for blind review>`

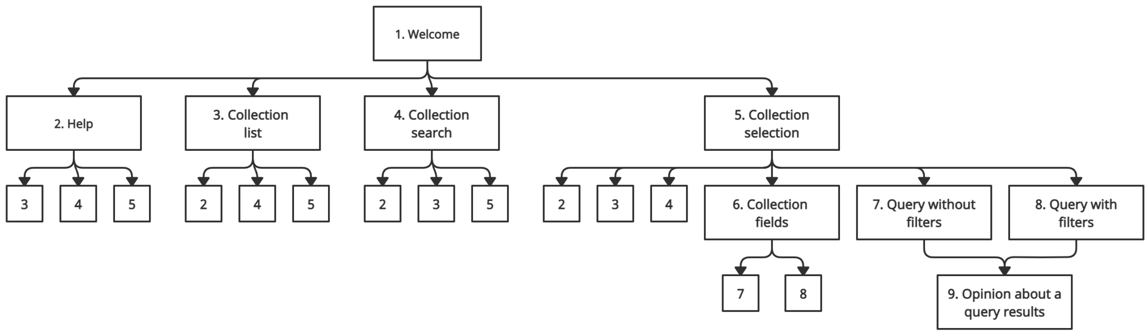

Fig. 4. Conversation flow of the developed chatbot.

ultimately presents them to the user. Specifically, the implemented routines are in charge of creating and launching SQL queries to a relational database that stores the OGD collections.

In the following, we describe the defined conversations intents, and present some of their triggering sentence patterns, as well as their corresponding SQL query templates.

(1) *Welcome*. This intent is triggered automatically at the beginning of a conversation. In it, the chabtot introduces itself, welcomes the user, and provides sentence examples for accessing to intent 2 (*help*).
(2) *Help*. In this intent, the chatbot provides a description and sentence examples of all the available intents. It is triggered when the user introduces sentences like "I need some advice," "Can you help me?" or simply "Help."
(3) *Collection list*. This intent is aimed to iteratively show the list of available collections in the dataset. It is triggered with sentences like "List me the data collections" and "What collections are available?"
(4) *Collection search*. This intent allows the user to perform a keyword-based search for retrieving collections that are indexed with certain terms (see Section 4.2 for more details). It is triggered with sentences like "Search for a collection" and "I am looking for a collection." The chabot responds to these sentences asking the user for the terms with which performing the search. Afterwards, it shows the titles of the retrieved collections. An example of conversation fragment for this intent is given in Figure 4 (left).
(5) *Collection selection*. In this intent, the user selects a particular collection on which searching for information of interest. Subsequently, the chatbot asks the user whether she wants to perform a query without (intent 7) or with (intent 8) filters. Sentences that trigger this intent are "I would like to use collection X," "I choose collection X" and "collection X." Executing this intent, the chatbot establishes the table to be used in the `FROM` clause of the SQL query to build.
(6) *Collection fields*. In this intent, the user asks for the fields (i.e., columns) of a selected collection (table). The chatbot lists the identifiers and names of the fields. Example sentences for this intent are "Available fields" and "Can you show me the list of columns of the collection?"
(7) *Query without filters*. This intent allows the user to launch an SQL query over the table of a selected collection without establishing filtering criteria, i.e., `WHERE` clauses.
    Within this intent, the chatbot asks the user for the fields of interest. The chosen fields (or all the fields of the table via the asterisk * parameter) are incorporated into the `SELECT` clause of the query.

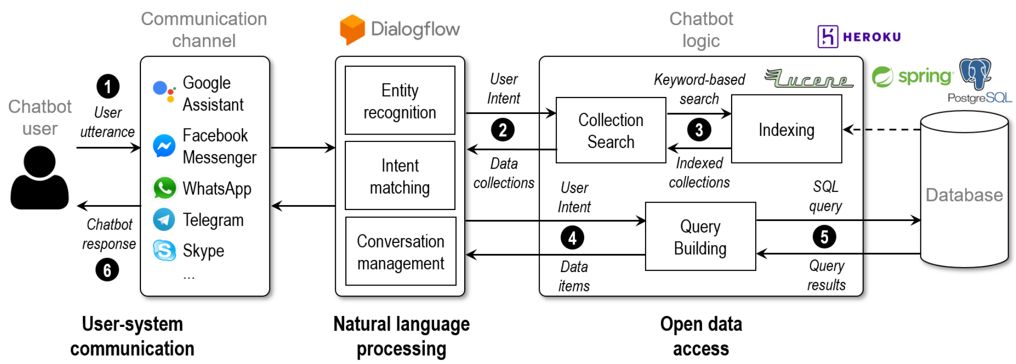

Fig. 5. Architecture of the developed chatbot.

Next, the chatbot asks the user if she wants to group the results by certain fields, and in affirmative case, it asks for such fields and the aggregation function (i.e., `SUM, MAX, MIN, AVG, COUNT`) that may be applied to the records of each group. This way, the chatbots gathers the parameters of the `GROUP BY` clause.
Finally, the user is allowed to state the fields (if any) by which sorting the query results (i.e., the `SORT BY` clause), and the maximum number (if any) of results to retrieve (i.e., the `LIMIT` clause).
With all the above, the query that is launched to the database has the following general form:

```
SELECT <fields1>, aggregationFunction(fieldX)
FROM  <table>
GROUP BY <fields2>
ORDER BY <fields3>
LIMIT <N>;
```

The chatbot shows the results of the query, and presents the user with two buttons to download the results in a CSV or Microsoft Excel[23] file, as shown in Figure 3 (right).

(8) *Query with filters*. This intent is equivalent to intent 7, but adding to the built query a `WHERE` clause with a number of filtering criteria. These criteria are of the form "`field operator value`", where `operator` is a relational operator (e.g., = for equal to, > for greater than, <= for less than or equal to, and != for not equal to). The allowed operators are described by the chatbot, as shown in Figure 3 (middle).

(9) *Opinion about a query results*. This intent is automatically triggered after query results are presented to the user, who is asked to provide feedback about them.

During a conversation, the chatbot keeps a log with the user's actions, queries and feedback, with their associated timestamps.

4.2 Architecture

Figure 5 depicts the architecture of our chatbot, which is built upon the Google DialogFlow framework. This framework allows easy and fast implementations of conversational systems. It enables i) the communication with a chatbot through a variety of commercial instant messaging and social networking services (e.g., Google Assistant, Facebook Messenger,

[23] Microsoft Excel, http://products.office.com/en-us/excel

WhatsApp, Telegram and Skype), ii) the processing of user utterances in natural language and the management of the user-chatbot conversations, and iii) the connection to external services and data sources.

The NLP modules apply state-of-the-art machine learning (deep network) techniques for two principal functionalities, namely *entity recognition* and *intent matching*. The former extracts from text (i.e., a user's utterance) entities such as dates, locations, proper nouns, and particular domain concepts. The latter identifies the intent underlying a user's utterance. Hence, with DialogFlow, a significant part of a chatbot implementation consists of the provision of example sentence patterns annotated with their concepts and intents. These annotated sentences are used by the framework to automatically build its machine learning models. For our chatbot, we defined a number of specific intents with associated annotated sentence patterns oriented to Open Data access (those above explained in Section 4.1). The considered entities and sentences were in Spanish. Their replacement by translations into other languages would allow using the chatbot in such languages.

The figure shows the components and modules of the chatbot, their interconnections as well as their data processes. The user accesses to and interacts with the chatbot via conversations on an instant messaging application (stage 1 in the figure). A user's textual utterance –manually written or automatically generated from voice messages– is sent to the NLP component of DialogFlow, which extracts entities and identifies the targeted intent.

If the intent is searching for a collection (stage 2), the chatbot asks the user for terms that may be associated to relevant collections, and uses the provided terms to launch a keyword-based search over a database index (stage 3) –created with tools such as Apache Lucene. As a result of the process, the user is presented with the list of OGD collections that have been indexed the input keywords[24] (stage 6). If the intent, in contrast, relates with some database operation (see Section 4.1), it is sent to a query building module (stage 4). A conversation is then maintained with the user, where she is iteratively asked for the elements needed to create a SQL query that represents the user's information needs. Once the query is created, it is launched to the database (stage 5), retrieving the data items of interest. These items are finally presented to the user in a suitable form (stage 6).

5 EXPERIMENTS

From the research literature surveyed in Section 2, only Neumaier et al.'s work [19] presents an evaluation of an OGD chatbot. In particular, it briefly reports a user study where (N=7) participants had to perform one search task (i.e., finding official statistics data from different countries concerning climate change) and to fill a questionnaire about system effectiveness and user expectation (acceptance).

To design our experiment, we considered previous works that evaluate chatbots in e-government (e.g., [3, 21]) and works that survey evaluation methodologies and metrics for conversational systems (e.g., [18, 20, 24]). In the next subsections, we describe the conducted evaluation methodology and the measured metrics.

5.1 Evaluation methodology

Our experiment consisted of a user study in a controlled setting. Participants were supervised and their actions were monitored and recorded. They conducted the study in personal computers with similar computing capabilities and internet connection conditions. All of them had some knowledge and expertise in managing spreadsheets –a common form of publishing OGD–, and only two had used SQL.

[24] The chatbot index handles keywords obtained from the OGD collections' titles, descriptions and field names.

Table 1. Items of the questionnaire oriented to evaluate public service values. They represent our proposed implementation of Makasi et al.'s theoretical framework presented in [16].

Id	Public service value	Questionnaire item
I1	Effectiveness	The system is capable of offering solutions to tasks as the proposed ones
I2	Efficiency	The system allows to quickly perform tasks as the proposed ones
I3	Openness	The system transparently discloses its identity and the rationale of its operation
I4	Fairness	The system conducts a fair process, without favoring or discriminating certain data
I5	Professionalism	The system interface follows a competent, respectful and consistent conduct
I6	Legitimacy	The system follows legal and reasonable procedures from the data handling and presentation point of view
I7	Accountability	The system offers explanation about its results, limitations and failures
I8	Trust	The system is reliable in terms of information management
I9	User orientation	The system allows the user to decide and control the actions to do
I10	Acceptability	The system is useful and beneficial for tasks as the proposed ones
I11	User satisfaction	The system increases satisfaction and interest in tasks as the proposed ones
I12	Privacy	The system preserves the user's privacy
I13	Adaptability	The system interface adapts to the used device, e.g., PC, tablet or mobile phone
I14	Collaborative intelligence	The system complements user skills and external services for tasks as the proposed ones

More specifically, a total of 12 people were individually recruited for our study. They voluntary agreed to participate and accept a consent form where they were informed about the experiment purpose and conditions. They were chosen in such a way that different age ranges and spreadsheet/SQL expertise levels were covered. In particular, they were 8 male and 4 female of ages ranging 18-24 years old (4), 25-34 years old (5), 35-44 years old (1), and 45-54 years old (2), with different education levels: secondary education (1), vocational education (3), Bachelor's degree (3), Master's degree (4), and Doctoral degree (1). Those with Higher Education levels had studied Sciences (1), Social Sciences (2), Arts and Humanities (1) and Engineering (4) careers. All of them used (keyword-based) web search engines frequently, and only one had not used a chatbot although he knew what a chatbot is. Only one had expertise with open data. Lastly, participants had diverse levels of knowledge/expertise on spreadsheets −low (2), medium (8) and high (2)− and databases/SQL −null (5), low (5) and high (2).

The study focused on the OGD collections freely available online in the Open Data portal of Madrid City Council. As far of January 2021, the catalogue of the portal contains 474 data collections, published in different formats (mostly CSV files and Microsoft Excel spreadsheets). Each collection has a title and a description of its inner structure, among other metadata.

We aimed to evaluate our chatbot in comparison to the traditional (cyclic) method followed to consume open data; that is, given an information need, a user i) searches for the relevant collection in an open data portal through keyword-based queries, ii) downloads the data file(s) of such collection, and utilizes an appropriate software to iii) open the file and iv) process its data until finding the answer to the information need. Hence, participants were split into two groups for searching and exploring open data collections: one group used the portal's search engine and Microsoft's Excel spreadsheet application, and the other group used our chatbot.

After filling a consent form and an anonymized, initial questionnaire to gather personal data (i.e., gender, age, level and area of education, degree of use of keyword-based search engines, and level of knowledge and expertise on spreadsheets and relational databases), participants received explanations about open data, OGD portals, and spreadsheets (or conversational systems).

Then, they had to perform three tasks of increasing difficulty:

- *Task 1: finding the public holidays in 2021.* The target collection was titled "work calendar" and contained the schedule of official working and non-working days from January 1, 2013 to December 31, 2021.

- *Task 2: finding the name and salary of the 10 councillors who earned the most money in 2020.* The target collection was titled "Remuneration of senior officials of Madrid City Council 2020" and contained the monthly salary of the city mayor, councillors and other senior managers of the city council in 2020.
- *Task 3: finding the total money allocated in the budget 2020 for culture activities.* The target collection was titled "General budget 2020" and contained detailed information about the expended budget of the city in 2020: execution date, city district, budget quantity, budget issue, category and chapter, to name a few.

The three tasks entailed performing queries whose terms were not keywords existing in the index of the portal search engine, so participants had to look for the appropriate synonym terms to find the collections. Once the relevant Excel files were retrieved, the tasks entailed different processing operations on the spreadsheet data to find the requested information. In particular, Task 1 entailed filtering the values of a couple of fields (columns); Task 2 entailed filtering by field, and applying an average computing function on a range of cells; and Task 3 entailed filtering by field, applying a summation computation function, and grouping records by the values of certain field.

For the two groups of participants, during the tasks, a variety of objective measures were recorded, such as the spent time, the number of performed queries and operations, and the task (non) completion. After finishing (or aborting) a task, participants were requested to fill an intermediate questionnaire to gather comments and subjective opinions about the perceived difficulty of the task and utility of the used tools, i.e., the portal search engine, the spreadsheet application, and the chatbot.

Finally, at the end of the experiment, participants were requested to fill a final questionnaire to gather their opinion about the degree of satisfaction of certain public service values of the used tools. This is detailed in the next section.

5.2 Evaluation metrics

As explained above, and differently to previous work, in our study we considered both objective and subjective metrics. On the one hand, all actions performed by participants on the systems (i.e., portal and chatbot) were recorded in log files, as done by the system developed by Ronzhin et al. [25]. The types, timestamps and results of such actions were stored so we could afterwards analyzed effectiveness and efficiency metrics, namely the correct/wrong performing or abortion of the tasks, the time spent on each task stage –namely, searching for the collection and obtaining the results from the collection–, and the numbers of search queries and data processing operations. The results and analysis of the log records are reported in Section 6.1. On the other hand, the intermediate questionnaires asked questions about the perceived degree of difficulty of the tasks, and the complexity of the system operations done in each task. A discussion of the received responses in the above questionnaries is given in Section 6.2.

In addition to these metrics, as a novel contribution of our work, we considered the theoretical framework recently proposed by Makasi et al. [16] aimed to establish the public values a chatbot-mediated public service should have. In particular, we implemented the framework as a final questionnaire with 14 items aimed to evaluate whether each considered OGD consumption method (portal+spreadsheets vs. chatbot) met the proposed public values. Table 1 shows the items of the developed questionnaire. As can be observed, the items are oriented to system (items I1 to I8), user (I9-I12) and contextual (I13, I14) aspects. The conclusions derived from the proposed questionnaire are presented in Section 6.3.

Table 2. Effectiveness and efficiency values achieved by participants with the portal and the chabot for the three tasks.

PORTAL

	Finding the collection					Obtaining the solution				
Task	% Completed correctly	% Completed wrongly	Avg. time	Avg. #queries	Avg. #collections	% Completed correctly	% Completed wrongly	Avg. time	Avg. #operations	Avg. total time
1	100.0	0.0	1.5	3.7	1.0	83.3	16.7	3.3	2.2	4.8
2	100.0	0.0	1.3	2.2	1.5	83.3	16.7	5.9	5.5	7.2
3	83.3	0.0	1.7	3.5	1.5	33.3	33.3	8.2	16.8	9.9
AVG.	**94.4**	**0.0**	**1.5**	**3.1**	**1.3**	**66.7**	**22.2**	**5.8**	**8.1**	**7.3**

CHATBOT

	Finding the collection					Obtaining the solution				
Task	% Completed correctly	% Completed wrongly	Avg. time	Avg. #queries	Avg. #collections	% Completed correctly	% Completed wrongly	Avg. time	Avg. #operations	Avg. total time
1	100.0	0.0	0.5	2.8	1.0	100.0	0.0	2.9	2.3	3.4
2	100.0	0.0	0.5	2.2	1.0	100.0	0.0	2.8	2.2	3.3
3	100.0	0.0	0.5	1.8	1.0	100.0	0.0	4.1	2.5	4.6
AVG.	**100.0**	**0.0**	**0.5**	**2.3**	**1.0**	**100.0**	**0.0**	**3.3**	**2.3**	**3.8**

6 RESULTS

As explained in Section 5.1, participants were split into two groups to perform the proposed three tasks. The first group was composed by those people who used the search engine of the OGD portal to retrieve collections and Microsoft Excel to find the solutions within the target collection, whereas the second group was formed by those people who only used the chatbot.

We next present and discuss the results of the study, distinguishing between these two groups of participants, analyzing objective measures of effectiveness and efficiency, and subjective measures of usability and public service values.

6.1 Effectiveness and efficiency

To measure and compare the achieved effectiveness and efficiency of the two evaluated systems –portal + spreadsheets vs. chatbot– we recorded all the actions performed by participants in the three tasks for both stages: finding the collection of interest and obtaining the requested solution from the collection. From our log records, we computed the percentage of uncompleted and (correctly or wrongly) completed tasks/stages as effectiveness metrics, and the average numbers of queries/operations and the average partial/total times as efficiency metrics.

Table 2 shows the values of these metrics. Regarding *effectiveness*, it can be observed that the chatbot allowed participants to correctly complete the three tasks. Using the portal and spreadsheets, in contrast, entailed that only 66.7% (22.2%) of the task attempts were successfully (wrongly) completed; 11.1% were not even completed. The improvement on effectiveness achieved by the chatbot was thus 33.3%.

With respect to *efficiency*, using the portal, there was an increasing time for performing the three tasks (4.8, 7.2 and 9.9 minutes), which is in accordance with their level of difficulty. This increment did not occur using the chatbot, with which the tasks were completed in similar times (3.4, 3.3 and 4.6 minutes). The third task took more time in both systems since it required grouping records and applying an aggregation function. In this context, it has to be highlighted the fact that the average time per task was reduced a 47.9% by the chatbot (from 7.3 to 3.8 minutes).

As shown in Table 2, the time decrease was achieved in the two task stages. On the one hand, participants had difficulties to find the relevant collections from the portal (on average, they spent 1.5 and 0.5 minutes with the portal and the chatbot, respectively), since they were not able to choose the appropriate keywords. The index of the portal

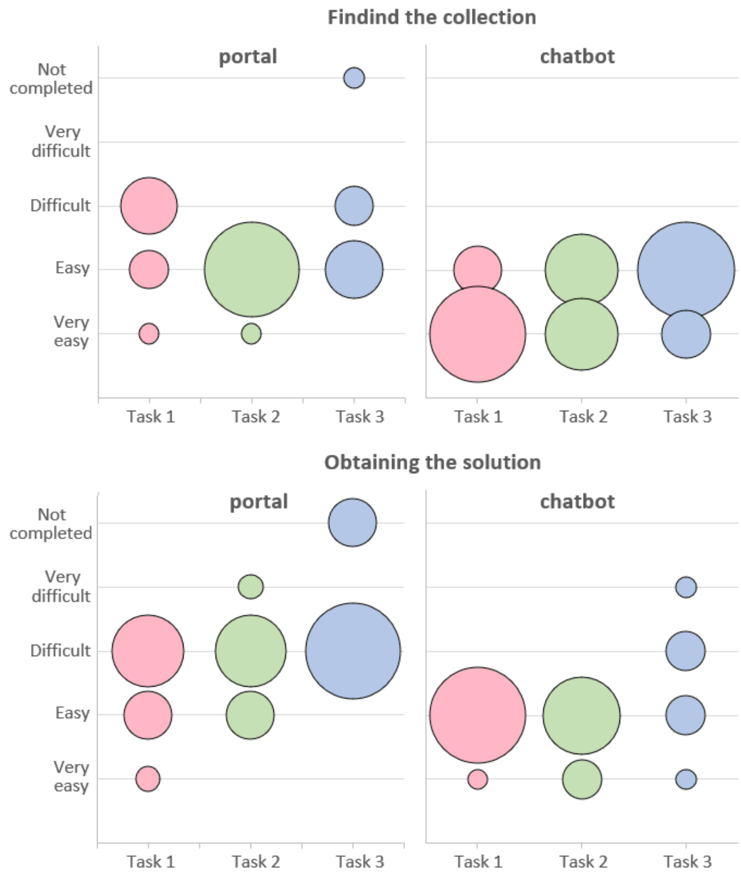

Fig. 6. Task difficulty levels perceived by participants.

search engine seemed to be more restricted than the chatbot index. This and the lack of query suggestions –e.g.,to correct misspellings or to replace terms by synonyms– were the limitations most commented by participants after the study. On the other hand, they spent much less time to obtain the solutions from the collections: on average, from 5.8 minutes by making operations over the spreadsheets to 3.3 minutes by conducting a quite stable number of iterations with the chatbot.

The level of expertise on spreadsheets helped participants to obtain the solutions from the Excel files more quickly: those with low, medium and high expertise spent 7.5, 5.5 and 4.2 minutes, respectively. The level of knowledge on SQL, in contrast, did not seem to influence on a more efficient use of the chatbot: participants with null, low and high knowledge spent 4.1, 2.7 and 3.2 minutes, respectively. This evidences the usability of the chatbot for users, regardless their expertise on relational databases and SQL. The same result applies to the participants' age, but could not be confirmed for their level of expertise on open data and chatbots due to the lack of enough sample data. Next, we analyze the participants' opinions about the system usability.

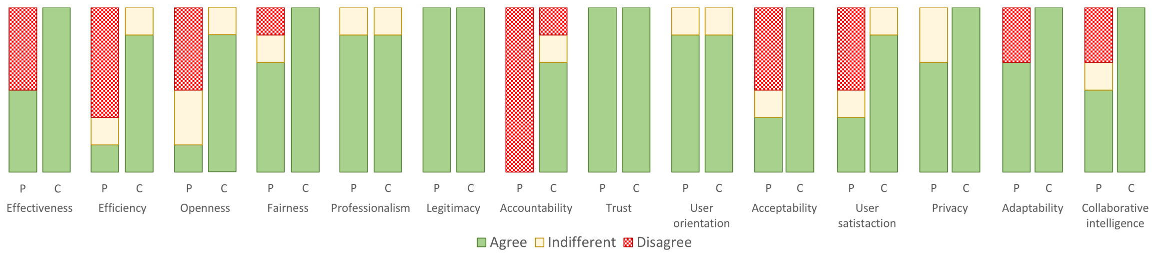

Fig. 7. Participants' opinion about the public service values provided by the portal (P) and the chatbot (C).

6.2 Usability

In the intermediate, task-oriented questionnaires, participants were asked about the difficulty of performing the three open data access tasks with the two evaluated systems. We next analyze the users' opinion on the tasks difficulty with respect to their complexity, i.e., Task 1 was the simplest one and Task 3 was the most complex one.

Figure 6 shows bubble charts representing the distribution of perceived levels of difficulty (vertical axis) of the three tasks (horizontal axis) and stages –finding the collection (charts at the top) and obtaining the results from the target collection (charts at the bottom)–, using the portal and spreadsheets (charts on the left) and using the chatbot (charts on the right).

It can be seen that in all settings, Task 3 was perceived as the most difficult task. It can be appreciated, nonetheless, that participants considered that finding the collection was easy or very easy using the chatbot in the three tasks. There was a significant percentage of participants considering that finding the collection in Tasks 1 and 3 had been difficult when using the portal search engine. This is in accordance with their number of queries done and time spent (cf. Section6.1 comments regarding the index limitations). A similar trend is observed for obtaining the solution from the target collection. However, in this stage, the levels of perceived difficulty were higher, even for the chatbot. This again is in accordance with the objective measures (cf. Section6.1), and the complexity of the grouping operation of Task 3.

In the intermediate questionnaires, participants were also asked about how intuitive the systems interfaces were. Analyzing their responses we did not find clear differences between the two systems. Participants agreed and commented that the interfaces (i.e., keyword-based search, spreadsheet manipulation, and chatbot interaction) are relatively easy to use with the appropriate knowledge and training. To further understand their opinion about the systems, in the next section, we look at their perception on the public service values offered by the systems.

6.3 Public service values

The final, system-oriented questionnaire was oriented to gather the participants' agreement with the provision of public (service) values by our open data access chatbot, in comparison to the open data portal. Figure 7 shows bar charts summarizing the received responses. For each of the 14 considered public values, two percentage chart bars are shown, associated to the portal (P) and the chatbot (C), and representing the percentages of *agree* (green bars), *indifferent* (yellow bars) and *disagree* (red, squared bars) responses. We next discuss those for which comments were given.

We start with the participants' opinion about the *effectiveness* and *efficiency* that can be achieved by the systems. As can be observed in the figure, there was a generalized agreement that the chatbot allowed to efficiently perform tasks as those proposed in the study. In contrast, less than 50% of the responses expressed such opinion for the portal. Several participants commented that in the portal case, the correct and fast completion of the data access task depends on the user's knowledge and expertise on the system (spreadsheet software), rather than on the system itself.

The chatbot clearly received more positive opinions than the portal in terms of *openness* and *accountability*. This, according to the participants' comments, was due to the help intent and the explanations given by the chatbot in the whole data access process. This also applies to the *fairness* value, for which certain participants expressed doubts about to the capability of the portal search engine to retrieve the collections related to the user queries.

All participants confirmed their *trust* on the veracity and *legitimacy* of the OGD used. They also agreed on the *adaptability* of the systems to the user device. However, some of them highlighted the fact that handling the spreadsheets in mobile phones would be too difficult or even not possible. Finally, there were some concerns on the *privacy* aspect by participants that raise doubts about whether and how web browsing cookies were being used in the portal.

7 CONCLUSIONS

This research is a first approach to using chatbots for improving access and appropriate use of government data. Nonetheless, the conducted preliminary user study has allowed us to find evidences about the benefits of the proposed chatbot to search and explore OGD with respect to traditional portals and data access methods.

Limitations of this research are mainly focused on the lack of different contexts and a small number of participants and tasks. Therefore, future research should be extended to a larger number of users, data collections and tasks, in less controlled scenarios. In this context, we aim to focus not only on non-expert citizens, but also on professionals, such as public and business managers, who have to perform policy- and decision-making actions based on the information underlying the OGD. With this goal in mind, we envision the need of incorporating data analysis and visualization functionalities into the chatbot. Moreover, we also aim to improve the chatbot to better and more clear address the considered public service values.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Science and Innovation (PID2019-108965GB-I00) and the Centre of Andalusian Studies (PR137/19). The authors thank to all people who participated in the reported study.

REFERENCES

[1] Aggeliki Androutsopoulou, Nikos Karacapilidis, Euripidis Loukis, and Yannis Charalabidis. 2019. Transforming the communication between citizens and government through AI-guided chatbots. *Government Information Quarterly* 36, 2 (2019), 358–367.

[2] Vito Walter Anelli, Tommaso Di Noia, Eugenio Di Sciascio, and Azzurra Ragone. 2019. Anna: A virtual assistant to interact with Puglia Digital Library (discussion paper). In *27th Italian Symposium on Advanced Database Systems*.

[3] Naomi Aoki. 2020. An experimental study of public trust in AI chatbots in the public sector. *Government Information Quarterly* 37, 4 (2020), 101490.

[4] Shubham Atreja, Pooja Aggarwal, Prateeti Mohapatra, Amol Dumrewal, Anwesh Basu, and Gargi B Dasgupta. 2018. Citicafe: An interactive interface for citizen engagement. In *23rd International Conference on Intelligent User Interfaces*. ACM, 617–628.

[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2011. Linked Data: The story so far. In *Semantic services, interoperability and web applications: Emerging concepts*. IGI Global, 205–227.

[6] Iván Cantador, María E Cortés-Cediel, and Miriam Fernández. 2020. Exploiting Open Data to analyze discussion and controversy in online citizen participation. *Information Processing & Management* 57, 5 (2020), 102301.

[7] Robert Dale. 2016. The return of the chatbots. *Natural Language Engineering* 22, 5 (2016), 811–817.

[8] Yingying Gao and Marijn Janssen. 2020. Generating value from government data using AI: An exploratory study. In *19th IFIP WG 8.5 International Conference on Electronic Government*. Springer, 319–331.

[9] Julien Hivon and Ryad Titah. 2017. Conceptualizing citizen participation in Open Data use at the city level. *Transforming Government: People, Process and Policy* 11, 1 (2017), 99–118.

[10] Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. 2012. Benefits, adoption barriers and myths of open data and open government. *Information Systems Management* 29, 4 (2012), 258–268.
[11] Sophia Keyner, Vadim Savenkov, and Svitlana Vakulenko. 2019. Open data chatbot. In *16th European Semantic Web Conference*. Springer, 111–115.
[12] Sara Lafia, Jingyi Xiao, Thomas Hervey, and Werner Kuhn. 2019. Talk of the town: Discovering open public data via voice assistants. In *14th International Conference on Spatial Information Theory*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 10:1–10:7.
[13] Daniel Lathrop and Laurel Ruma. 2010. *Open government: Collaboration, transparency, and participation in practice*. O'Reilly Media, Inc.
[14] Andreas Lommatzsch. 2018. A next generation chatbot-framework for the public administration. In *18th International Conference on Innovations for Community Services*. Springer, 127–141.
[15] Ann Macintosh. 2008. E-democracy and e-participation research in Europe. In *Digital government*. Springer, 85–102.
[16] Tendai Makasi, Alireza Nili, Kevin Desouza, and Mary Tate. 2020. Chatbot-mediated public service delivery. *First Monday* 12, 25 (2020).
[17] Stavros Mallios and Nikolaos Bourbakis. 2016. A survey on human machine dialogue systems. In *Prodeedings of the 2016 7th International Conference on Information, Intelligence, Systems & Applications*. IEEE, 1–7.
[18] Wari Maroengsit, Thanarath Piyakulpinyo, Korawat Phonyiam, Suporn Pongnumkul, Pimwadee Chaovalit, and Thanaruk Theeramunkong. 2019. A survey on evaluation methods for chatbots. In *2019 7th International Conference on Information and Education Technology*. ACM, 111–119.
[19] Sebastian Neumaier, Vadim Savenkov, and Svitlana Vakulenko. 2017. Talking open data. In *14th European Semantic Web Conference*. Springer, 132–136.
[20] Dijana Peras. 2018. Chatbot evaluation metrics. In *36th International Scientific Conference on Economic and Social Development*. Varazdin Development and Entrepreneurship Agency, 89–97.
[21] Yulia Petriv, Regina Erlenheim, Valentyna Tsap, Ingrid Pappel, and Dirk Draheim. 2019. Designing Effective Chatbot Solutions for the Public Sector: A Case Study from Ukraine. In *6th International Conference on Electronic Governance and Open Society: Challenges in Eurasia*. Springer, 320–335.
[22] Simone Porreca, Francesco Leotta, Massimo Mecella, and Tiziana Catarci. 2017. Chatbots as a novel access method for government open data. In *25th Italian Symposium on Advanced Database Systems*. CEUR Workshops, 122–129.
[23] Simone Porreca, Francesco Leotta, Massimo Mecella, Stavros Vassos, and Tiziana Catarci. 2017. Accessing government open data through chatbots. In *17th International Conference on Web Engineering*. Springer, 156–165.
[24] Ranci Ren, John W Castro, Silvia T Acuña, and Juan de Lara. 2019. Evaluation Techniques for Chatbot Usability: A Systematic Mapping Study. *Intl. Journal of Software Engineering and Knowledge Engineering* 29, 11n12 (2019), 1673–1702.
[25] Stanislav Ronzhin, Erwin Folmer, Pano Maria, Marco Brattinga, Wouter Beek, Rob Lemmens, and Rein van't Veer. 2019. Kadaster knowledge graph: Beyond the fifth star of open data. *Information* 10, 10 (2019), 310.
[26] Navid Tavanapour, Mathis Poser, and Eva AC Bittner. 2019. Supporting the idea generation process in citizen participation - Toward an interactive system with a conversational agent as facilitator. In *27th European Conference on Information Systems*. Association for Information Systems.
[27] Camilo Thorne. 2017. Chatbots for troubleshooting: A survey. *Language and Linguistics Compass* 11, 10 (2017), e12253.
[28] Vishanth Weerakkody, Zahir Irani, Kawal Kapoor, Uthayasankar Sivarajah, and Yogesh K Dwivedi. 2017. Open data and its usability: an empirical view from the Citizen's perspective. *Information Systems Frontiers* 19, 2 (2017), 285–300.
[29] Matthew M Young. 2020. Implementation of Digital-Era Governance: The Case of Open Data in US Cities. *Public Administration Review* 80, 2 (2020), 305–315.