A Voice-based Travel Recommendation System Using Linked Open Data

Krzysztof Kutt[1][0000−0001−5453−9763], Sebastian Skoczeń[2][0000−0003−0242−2373], and Grzegorz J. Nalepa[1,2][0000−0002−8182−4225]

[1] Jagiellonian Human-Centered Artificial Intelligence Laboratory (JAHCAI) and Institute of Applied Computer Science, Jagiellonian University, 31-007 Kraków, Poland
krzysztof.kutt@uj.edu.pl, gjn@gjn.re
[2] AGH University of Science and Technology, 30-059 Kraków, Poland

Abstract. We introduce J.A.N.E. – a proof-of-concept voice-based travel assistant. It is an attempt to show how to handle increasingly complex user queries against the web while balancing between an intuitive user interface and a proper knowledge quality level. As the use case, the search for travel directions based on user preferences regarding cuisine, art and activities was chosen. The system integrates knowledge from several sources, including Wikidata, LinkedGeoData and OpenWeatherMap. The voice interaction with the user is built on the Amazon Alexa platform. A system architecture description is supplemented by the discussion about the motivation and requirements for such complex assistants.

Keywords: Voice assistant·Human-computer interaction·Linked Open Data.

1 Introduction and Motivation

Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant – are the names of the most popular voice assistants. They are software agents that provide an interface between the user and the device, e.g., a smartphone, a PC or a stand-alone home appliance. With their help, one can send text messages, set calendar entries or control media playback using only own voice [5]. As they are becoming more popular, more problems appear. The failure of Microsoft's *Tay* showed us that as a society we are not ready yet for loose conversations with an electronic agent. Both from the design point of view, where unpredictable user behaviour has to be predicted, as well as unrealistic user expectations for such systems [9]. Additionally, it is important to pay attention to the problems of automatic processing of knowledge in general-purpose assistants. An example is Google's Assistant, which believes that the horse has 6 legs [10].

To overcome problems related to the quality of knowledge, it is better to focus on creating modules responsible for smaller, but more complicated tasks. Data available in the Semantic Web and the whole stack of Semantic Web technologies may provide a reliable architecture for knowledge representation and reasoning [1]. In particular, the projects grouped in the Linked Open Data (LOD)[3] community may be useful. They

[3] See: https://lod-cloud.net/.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31

are publicly accessible for free, without any licensing, that encourages collaboration between people and boost the development of knowledge graphs [3]. The strength of the LOD is the use of URIs as identifiers, thanks to which we can easily combine and transform data from many different sets, resulting in a so-called *semantic mashup* [7]. Such mashups were also created for travel-related tasks (e.g., [4]), but they were not combined with other data sources nor with a voice interface.

In this paper, the J.A.N.E. voice assistant is introduced. In contrast to general-purpose assistants, J.A.N.E. is prepared for specific applications using reliable sources of knowledge and the interaction is conducted in a loose but very controlled manner. Our second motivation was to show the promising capabilities of a hybrid system that combines (a) standard methods of search, (b) powerful semantic queries, and (c) voice-based user interface to process complex time-consuming queries in a natural way. This is achieved by creating a solution based on Amazon's Alexa platform that will be capable of recommending the best travel destination based on user's preferences.

The rest of paper is organized as follows. In Sect. 2 we discuss the specification and the workflow of the tool. Then, four datasets used by J.A.N.E. to perform its tasks are described in more detail in Sect. 3. The evaluation is provided in Sect. 4. The paper is concluded in Sect. 5.

2 J.A.N.E. Overview

J.A.N.E. (Just Another Neat Expeditionist) name has been chosen to explicitly show the focus area of this voice assistant and to make the whole system more "human". It was also taken into consideration that female voice assistants are better perceived by users and are considered more trustworthy [8]. Finally, this approach encourages end-users to interact with the system in a more natural way.

The goal of J.A.N.E. is to seek travel destinations based on user preferences. The system takes into consideration the initial point of the journey, all the destinations available via a direct flight, the weather in the final location, on top of which the alignment with user preferences is checked. Three specific use cases are defined as a workbench:

Cuisine: query all of the possible locations for restaurants, cafés and bars that aligns with the requested type of cuisine, e.g., Italian, Asian, French,

Art: choose a destination based on art collection that can be found in local galleries and museums, and

Activities: search for possible sport activities in the destination location, e.g., riding a bike, swimming, ice skating.

J.A.N.E. is built on Alexa, as it is a part of well-established stack of Amazon Web Services (AWS) technologies. The system consists of three main parts (see Fig. 1):

1. The Alexa Echo Dot Gen. 2 device, a hardware interface for the user, supported by the voice front-end interface developed with Alexa Voice Services (AVS),
2. The AWS Lambda back-end, responsible for processing the request and querying the data sources,
3. Four services (DynamoDB, Wikidata, LinkedGeoData, OpenWeatherMap) providing the actual data (see Sect. 3).

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31

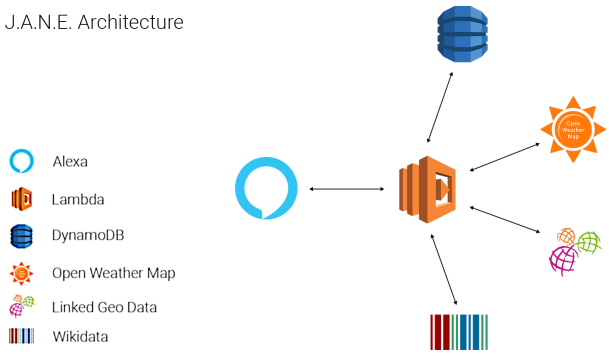

Fig. 1. J.A.N.E. Architecture Overview

After the user asks a question, J.A.N.E. takes a few steps to generate the answer:

1. First, the user voice request is parsed by the AVS accordingly to the defined interaction model (see Listing 1.1). It consists of distinguishable utterances that are used as schemes to process the voice.
2. Then, generated JSON is sent to AWS Lambda back-end, and the user is informed about the current state (e.g., *Give me a while... I'll do a small research for you*).
3. All available connections from the current location are checked using DynamoDB database (see Sect. 3.1).
4. The user preferences are transformed into SPARQL queries to search for matches in all of the previously discovered locations (see Sect. 3.2–3.3).
5. The locations that are fulfilling users requirements along with the number of matches are then passed through the forecast check (see Sect. 3.4).
6. Finally, the destination with the highest score is chosen and delivered as a voice response to the user.

3 Data Sources

3.1 Amazon DynamoDB Database

As there are no reliable LOD services with flight connections data, two database dumps were used: one containing airport details and the other containing information about all routes operated by the airlines in January 2012. The final dataset containing four columns: *Airport IATA Code*, *Municipality*, *Coordinates* and *Destinations* was stored in the DynamoDB.

The current location of the user is used for a simple database lookup. It is given either as the city name (specified by the user) or as the GPS coordinates from the device.

3.2 Wikidata Query Service

Wikidata[4] is a database that contains structured data aimed at support for other wiki projects such as Wikipedia and Wikivoyage. Access to the data is free of charge and the dataset is developed by both human editors and automated bots [12].

[4] See: https://www.wikidata.org.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31

In Listing 1.1 one can notice that the interaction model concerning art includes questions (specified in `samples` element) about specific painters or artistic movements. When such a statement is recognized, the corresponding excerpt from the user's input is stored in one of the defined slots: `artMovement` or `painter`. For both types, separate Wikidata queries are prepared to obtain relevant information (see Listing 1.2). Both are processed using the https://query.wikidata.org/sparql?query= endpoint.

3.3 LinkedGeoData SPARQL Endpoint

LinkedGeoData is a project that uses the information collected by the OpenStreetMap and transforms it into RDF triples [11]. J.A.N.E. uses this resource particularly to handle cuisine and activities use cases. A sample query for counting all gastronomy locals in a particular location can be seen in Listing 1.3. Activities are pre-processed before being placed in the corresponding placeholder in a query, e.g., if a user is searching for "swim" or "dive", `SwimmingPool`, `WaterPark`, `DiveCenter` and `Beach` are added to the activity list.

3.4 OpenWeatherMap API

OpenWeatherMap[5] is a free web service that provides weather forecast. It can be queried using the endpoint: http://api.openweathermap.org/data/2.5/forecast?lat=_LAT_&lon=_LON_&appid=_APIKEY_. The result consists of 40 weather records that represent the weather state every 3 hours for the upcoming 5 days. Every record is evaluated and the points are cumulated into a weather score, in the manner presented in Tab. 1. The final score is then calculated as: $(1+weatherScore)\cdot matchScore$.

4 Evaluation

Because one of the aims of this project is to provide the user with the answer in less than 120 seconds, the following 3 approaches were tested, using a "cuisine" intent query launched on 10 biggest cities in Europe, to find the quickest solution:

Querying all destinations with a single query: the query held all 10 cities in a form of an array consisting of the city name, latitude and longitude: `VALUES ( ?city ?latt ?long) {( "saint petersburg" 59.9343 30.3351), ...}`. This query took *56 s* on average to complete the task.

Querying all destinations one after another: the queries were launched one after another. Surprisingly, the task was finished in average time of *49 s*, which is 7 s on average faster than performing a joint query.

Querying all destinations in parallel: the queries were launched in parallel (respecting the endpoint limitations), not waiting for the previous one to finish. In this case the task was completed in *9 s* on average.

Furthermore, a side effect was observed. While querying endpoints with the same query multiple times, the performance boost was visible and the query was executed in the lower amount of time until it reached the lowest possible time. This fact might be used to establish the connection with endpoints before they are actually queried.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31

```
{ "interactionModel": {
    "languageModel": {
      "invocationName": "jane",
      "intents": [ {
          "name": "art",
          "slots": [
            { "name": "artMovement", "type": "AMAZON.SearchQuery" },
            { "name": "painter", "type": "AMAZON.Author" }
          ],
          "samples": [
            "where should I travel to discover {painter} works",
            ...
            "where I should travel to discover {artMovement} movement",
            "where I should travel to learn more about {artMovement}"
          ]
        }, ... ], } } }
```

Listing 1.1. An excerpt from the Alexa interaction model concerning art.

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <https://www.wikidata.org/wiki/Property:>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>

SELECT ?coord (COUNT(*) as ?count)
WHERE {
  ?painting wdt:P31 wd:Q3305213 ;      ## instance of (P31) painting (Q3305213)
            wdt:P276 ?location ;       ## location (P276)
            wdt:P135 wd:_USERINPUT_ . ## movement (P135)
  ?location wdt:P625 ?coord .          ## coordinate location (P625)
  SERVICE wikibase:label {  bd:serviceParam wikibase:language "en"  }
}
GROUP BY ?coord
ORDER BY DESC(?count)
```

Listing 1.2. SPARQL query against Wikidata to search and count all places related to art movement processed from user input (`_USERINPUT_` placeholder). The creator property (`P170`), instead of the movement (`P135`), is used for `painter`-related query.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX lgdo: <http://linkedgeodata.org/ontology/>
PREFIX ogc: <http://www.opengis.net/ont/geosparql#>
PREFIX geom: <http://geovocab.org/geometry#>

SELECT ?city ?latt ?long (COUNT(?name) AS ?count)
WHERE { VALUES (?city ?latt ?long) { ( _MUNICIPALITY_ _LAT_ _LON_ ) }.
    ?instance a ?gastronomy_local ;
        rdfs:label ?name ;
        geom:geometry [ ogc:asWKT ?entityGeo ] .
  FILTER (bif:st_intersects (?entityGeo, bif:st_point(?long, ?latt), 20)).
  FILTER (?gastronomy_local = lgdo:Restaurant ||
          ?gastronomy_local = lgdo:Cafe ||
          ?gastronomy_local=lgdo:Bar).
}
```

Listing 1.3. GeoSPARQL query against LinkedGeoData to count all gastronomical locals in the radius of 20 km from specific location. `_MUNICIPALITY_`, `_LAT_` and `_LON_` are placeholders replaced with details from DynamoDB.

Table 1. Weather records evaluation scheme.

Weather name:	Clear	Cloudy	Foggy	Drizzle	Snowy	Rainy	Thunderstorm
Points:	+0.02	+0.01	−0.002	−0.004	−0.006	−0.0125	−0.02

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31

To compare J.A.N.E. performance with the performance of real users, a simplified use case has been made based on "artist" intent. All of the subjects were asked to manually find a destination that fulfills two requirements: 1) It is possible to fly there directly from Cracow, Poland, 2) There is the biggest collection of Vincent Van Gogh paintings in the world. The participants could use any website they wanted along with the device that they felt the most comfortable with to simulate normal conditions of such research (e.g. google search engine, chrome web browser and windows operating system). In this experiment participants weren't asked to check the weather, check other possible locations, calculate how many matching paintings are there and then calculate the final score picking up the most promising location. It is worth noting, that these are tasks J.A.N.E. is performing every time when handling a user's request.

The total of 30 participants (19–56 years old) took part in the experiment. It took *153 s* on average for the participants to search the query[6], whereas J.A.N.E. needed *13 s* to find the solution performing all of the additional processing. The results of this experiment are very promising showing that such system is almost 12 times faster than a regular user performing manual research. Moreover, the superiority of such travel recommendation system over manual research not only lies in the speed but also in the precision of the results giving here an exact number of the matching entities and providing a recommendation that also takes into account the latest weather forecast.

5 Discussion

J.A.N.E.—the voice assistant presented in this paper—provides an intermediary layer between user voice requests (gathered with Alexa platform) and the data stored in the Semantic Web. The system is a hybrid solution using data from both standard databases and Open Linked Data, gathering user preferences, interpreting their intents and returning the desired response using a synthesised voice.

The presented solution is in line with current trends in the e-Tourism, where attention is drawn to the need for voice communication in natural language and the integration of various sources of knowledge. Other use cases developed in this area include, e.g., room booking and conversation about local attractions (see [2] for overview on current issues in knowledge-based assistants in e-Tourism).

This study has some potential limitations. First of all, only three use cases of seeking flights based on user preferences were introduced as a proof-of-concept to show the possibility of using a vocal assistant with heterogeneous domains. We assume, that the tool has the potential to be integrated (and extended) using different external knowledge sources. Secondly, currently used voice-to-SPARQL mechanism may be improved with the use of recent automatic state-of-the-art methods [6,13]. However, it should be noted that, unlike these methods, we also query various APIs and classic SQL databases, and not just SPARQL endpoints. Finally, the workflow and the queries could be optimized so that the answer for the user will be generated faster.

[4] See: http://linkedgeodata.org.
[5] See: https://openweathermap.org/.
[6] There were some participants that knew the answer and they had significantly lower result from the average.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31

We assume that the presented approach can have a wide range of applications everywhere, where a user-friendly endpoint is needed to access well-defined data. Considering the use cases implemented so far, the system could be extended with modules checking available accommodation, the price of the ticket or even if there are no other overlapping events in users' personal calendar. The system can also be enriched with the ability to learn the user priorities, e.g., weather over the number of museums, providing more precise results over time. Finally, we plan to examine how people are using a voice-based travel assistants and identify the limitations in current solutions that can be addressed via knowledge-based reasoning. One of these deficiencies may be a mechanism to clarify ambiguities, e.g., to determine whether Venice in Italy or Poland[7] is in question.

References

1. Allemang, D., Hendler, J.A.: Semantic Web for the Working Ontologist - Effective Modeling in RDFS and OWL, Second Edition. Morgan Kaufmann (2011)
2. Angele, K., Fensel, D., Huaman, E., Kärle, E., Panasiuk, O., Şimşek, U., Toma, I., Wahler, A.: Semantic Web Empowered E-Tourism, pp. 1–46. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-05324-6_22-1
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. **5**(3), 1–22 (2009). https://doi.org/10.4018/jswis.2009081901
4. Cano, A.E., Dadzie, A.S., Ciravegna, F.: Travel mashups. In: Endres-Niggemeyer, B. (ed.) Semantic Mashups: Intelligent Reuse of Web Resources, pp. 321–347. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36403-7_11
5. Hoy, M.B.: Alexa, siri, cortana, and more: An introduction to voice assistants. Medical Reference Services Quarterly **37**(1), 81–88 (2018). https://doi.org/10.1080/02763869.2018.1404391, pMID: 29327988
6. Jung, H., Kim, W.: Automated conversion from natural language query to SPARQL query. Journal of Intelligent Information Systems **55**(3), 501–520 (2020). https://doi.org/10.1007/s10844-019-00589-2
7. Malki, A., Benslimane, S.M.: Building semantic mashup. In: ICWIT. pp. 40–49 (2012)
8. Mitchell, W.J., Ho, C.C., Patel, H., MacDorman, K.F.: Does social desirability bias favor humans? explicit–implicit evaluations of synthesized speech support a new HCI model of impression management. Computers in Human Behavior **27**(1), 402–412 (2011). https://doi.org/10.1016/j.chb.2010.09.002
9. Neff, G., Nagy, P.: Talking to bots: Symbiotic agency and the case of tay. International Journal of Communication **10**, 4915–4931 (2016)
10. Schwartz, B.: Google works to fix how many legs horses & snakes have (Apr 2019), https://www.seroundtable.com/google-how-many-legs-horses-snakes-27491.html
11. Stadler, C., Lehmann, J., Höffner, K., Auer, S.: Linkedgeodata: A core for a web of spatial open data. Semantic Web **3**(4), 333–354 (2012). https://doi.org/10.3233/SW-2011-0052
12. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledge base. Communications of the ACM **57**, 78–85 (2014), http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext
13. Yin, X., Gromann, D., Rudolph, S.: Neural machine translating from natural language to sparql. Future Generation Computer Systems **117**, 510–519 (2021). https://doi.org/10.1016/j.future.2020.12.013

[7] See https://en.wikipedia.org/wiki/Wenecja for the polish village called Venice.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77967-2_31