Jarvis: A Voice-based Context-as-a-Service Mobile Tool for a Smart Home Environment

Ngoc Dung Huynh, Mohamed Reda Bouadjenek, Ali Hassani, Imran Razzak, Kevin Lee,
Chetan Arora, Arkady Zaslavsky
School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, VIC 3216, Australia
E-mails: {ndhuynh, reda.bouadjenek, ali.hassani, imran.razzak, chetan.arora, arkady.zaslavsky}@deakin.edu.au

***Abstract*—In this paper we introduce Jarvis, a context-as-a-service mobile tool, which enables context-aware data collection, service discovery, and computer-aided situational awareness through a conversational User Interface (UI). At the core of Jarvis are two main components: (i) a voice-based UI to translate speech to Context Definition and Query Language (Speech-to-CDQL), and (ii) an operational component called Context-as-a-Service (CoaaS), which enables smart things and IoT silos to discover, validate and share relevant and dependable context. The UI is based on two machine learning models: a Speech-to-Text model and a Text-to-CDQL model based on an encoder-decoder architecture. Jarvis is developed as a mobile application that allows people with different backgrounds to interact with various IoT devices. Our demo shows how easy Jarvis can be used for context-aware data collection and to interact with diverse objects in a smart home environment through voice.**

***Index Terms*—Internet of Things (IoT), Context Definition and Query Language (CDQL), Smart Home.**

I. Introduction

The usage of Internet of Things (IoT) devices by both, individuals and organizations has been increasing rapidly over the last decade, and it is widely expected that the use of IoT will continue to expand rapidly [1]. An IoT device is a physical device connected to the internet that collects, shares, or uses data [2]. The smart home is one of the most popular IoT applications and has been a hot research topic for a while. In particular, a smart home is an environment that uses the latest electronic technologies and computerization to control or automate a number of different aspects of the house, such as temperature regulation, lighting, and to track energy consumption [3]. These smart devices can be remotely controlled through smartphones, tablets, or computers (laptops) [4].

Collecting data generated by IoT devices follows the three primary principles of Big Data [5]: speed, volume, and variety. This leads to big challenges to store, transform, and analyze the data generated for optimal resource usage. As a result, relational databases are becoming more and more popular in storing, managing, updating, and extracting data for IoT development. However, these databases require an understanding of the query languages for updating and searching data content from the host databases. Popular query languages such as SQL, AQL, SPARQL, Datalog, and DMX can be used. However, these query languages do not effectively work with contextual data commonly found in IoT applications.

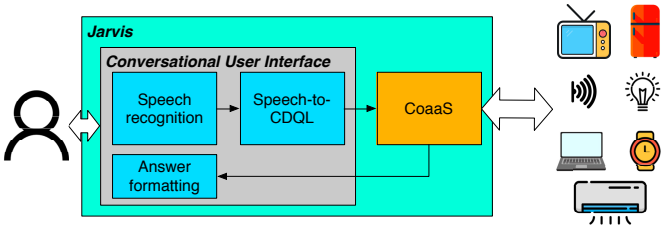

Fig. 1: Architecture overview of Jarvis.

In this paper, we introduce Jarvis[1], a context-as-a- service mobile application to enables context-aware data collection, service discovery, and computer-aided situational awareness. Jarvis relies on the Context Definition and Query Language (CDQL) introduced by Hassani et al. [6], which is an advanced query language that allows smart devices in an IoT system to exchange, reuse, and share data. This approach has two main components: a Context Definition Model and a Context Query Language. The former aims to define the situation and level content, while the latter is a query language that asserts contextual information requirements without understanding the details of the data structures.

Because CDQL requires an understanding of the schema of the database and the roles of the entities in the query, only an expert with advanced knowledge in CDQL can effectively exploit it. Therefore, Jarvis relies on a UI that allows to converse with the system and to convert speech to CDQL to query the CoaaS component and thus, to interact with diverse objects in a smart home environment. To the best of our knowledge, Jarvis is the first tool that enables context-aware data collection, service discovery, and computer-aided situational awareness through a conversational UI.

II. Architecture Overview

Figure 1 describes the logical architecture of our tool, Jarvis, a context-as-a-service mobile tool with a voice-based UI. At the core of Jarvis are two components that we now describe.

A. Conversational User Interface (UI)

This component allows the user to interact and converse with Jarvis through voice or speech commands. The responsibility of the UI is to take the human voice question and

[1]https://github.com/parkerhuynh/IEDG

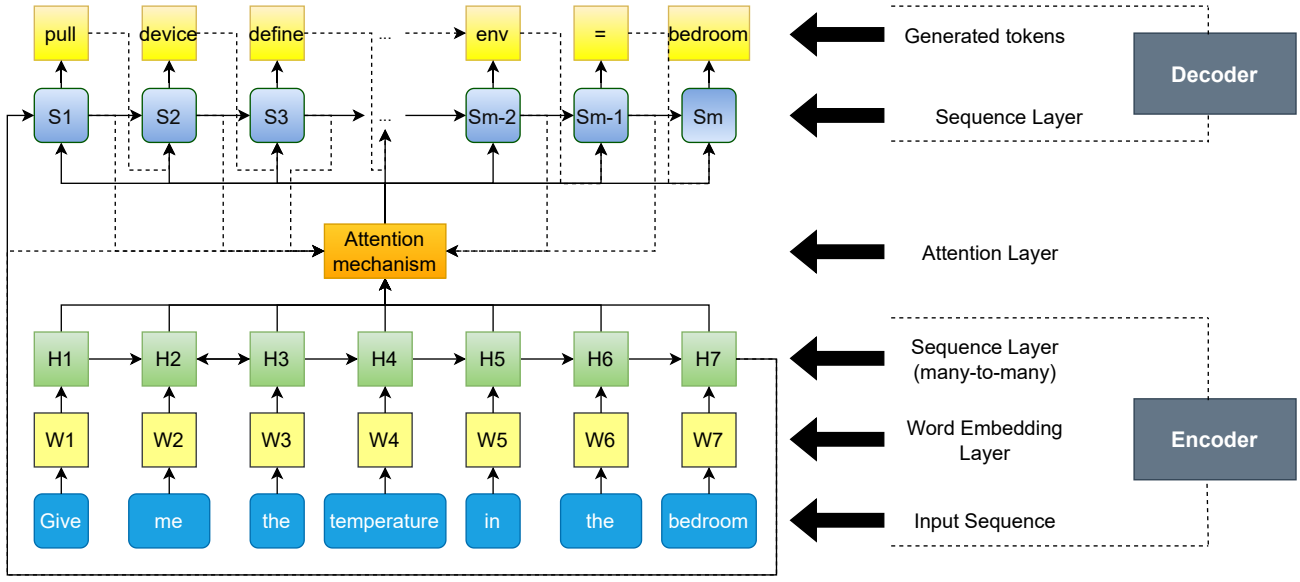

Fig. 2: Attention Encoder-Decoder Architecture.

transform it into the most likely CDQL query. After that, it will send the output to the CooaS platform via a POST request.

Specifically, the UI contains 2 machine learning models: (i) a speech recognition model that aims to convert human speech into text transcription, and (ii) a Text-to-CDQL model, which takes the text from the speech recognition model and generates the corresponding CDQL. We provide a detailed description of each model in the following.

Automatic Speech Recognition: For the automatic speech recognition component, we use an existing model called *Listen, Attend, and Spell* introduced by Google [7]. This model is created based on the attention encoder-decoder architecture that originates from translation work. *Listen* is the encoder, *Attend* is the attention mechanism, and the spell is the decoder. This model reaches a Word Error Rate of about 10.3%.

Text-to-CDQL: The Text-to-CDQL system is based on an encoder-decoder RNN-based model [8], [9]. This architecture aims to resolve the sequence-to-sequence nature of machine translation, where the input sequence differs in length from the output sequence. The Encoder-Decoder model has two main parts: Encoder and Decoders. The encoder is an RNN that takes the input sequence and converts it into hidden states.

In particular, the Text-to-CDQL component relies on an Attention-based Neural Machine Translation model, which has been introduced to improve the basic Encoder-Decoder Machine Translation [10], [11]. The decoder uses attention to focus on parts of the hidden states of the encoder selectively. The attention takes a sequence of vectors as input for each word and returns an "attention" vector. The "attention" vector provides a weighted context from the encoder to the decoder. The decoder will pay more "attention" to some tokens than others when predicting outputs at each step (Figure 2).

A beam search [12] is used to look for the top 3 most likely CDQL sequences. The CDQL sequences which are not valid according to a finite automaton that checks the CDQL syntax are filtered out during the beam searching process. The most likely sequence as a result of the CDQL-aware beam search is selected as the corresponding CDQL.

B. *CooaS platform*

CooaS is a context management middle software created to operationalize context-awareness in the IoT domain. Users can send requests to the platform to extract and update context or contextual data about IoT entities [6]. This component serves 2 purposes: First, it stores the data generated from the IoT simulator into the database. Second, it receives the CDQL query from the UI, executes the query, and returns the result to the UI.

III. What will be demonstrated?

The demonstration will be illustrated using a scenario related to controlling various objects of a smart home. We simulate a smart home that includes several environment sensors or smart devices such as temperature sensors and smart TVs in different part of the home, like the bedrooms and the living room. The entire simulated smart home environment is represented in Figure 3. We use a tool called IoT-Data-Simulator provided by IBA Group to simulate the smart home[2]. The data from simulated devices is sent to the data flatform CooaS via POST requests.

In this demonstration, a user can converse with Jarvis as described in the following scenarios.

A. *Basic scenario*

In the **base scenario**, a user asks about one measurement in a specific area of the smart home in Figure 3. Here, the

[2]https://github.com/IBA-Group-IT/IoT-data-simulator

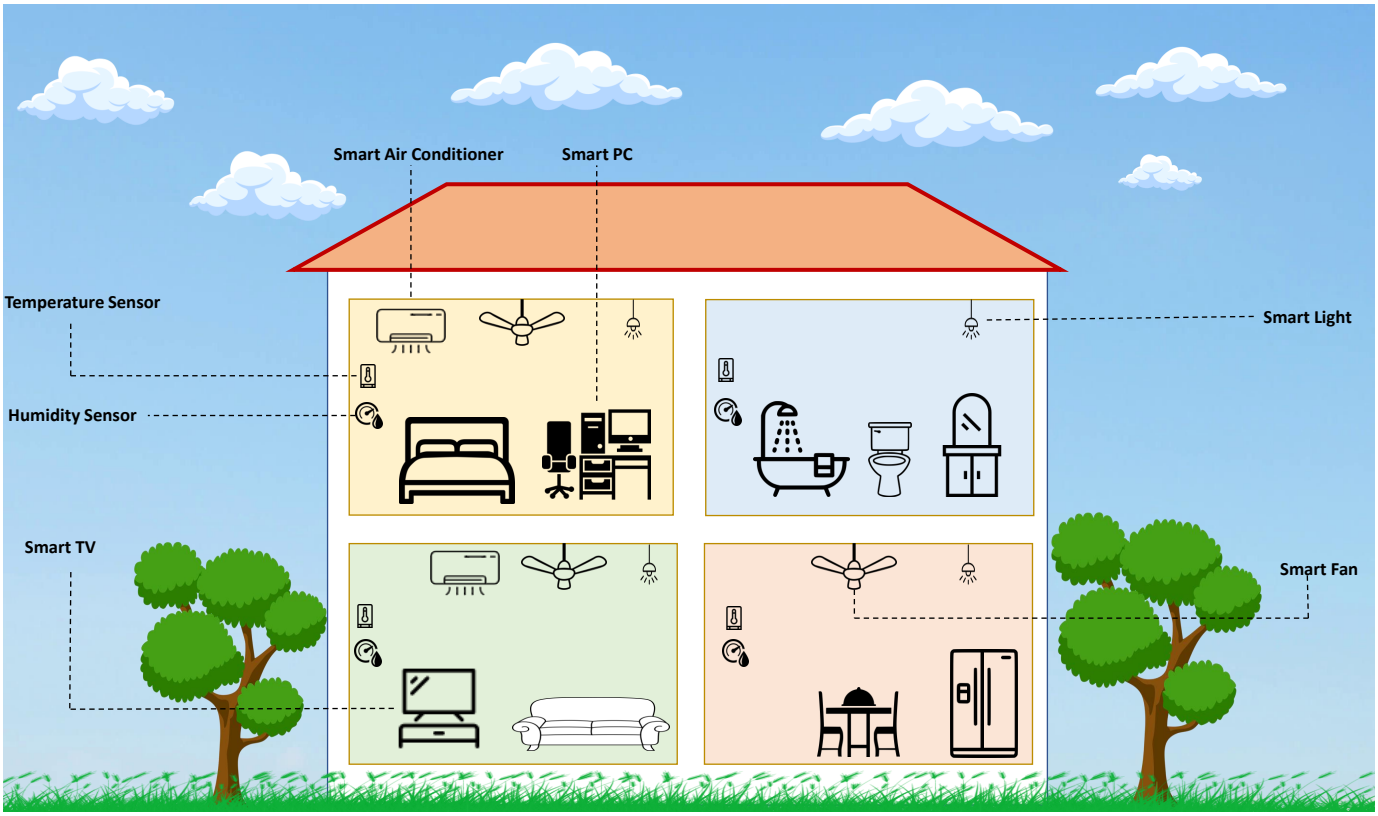

Fig. 3: Simulated Smart Home for the demonstration.

(a) (b) (c)

Fig. 4: Examples of basic scenarios that will be demonstrated: (a) ask about one measure in a specific room of the smart home, (b) ask about the state of one smart device in a specific room, and (c) ask about the state of the TV in the living room.

user is fully informed about the data he wants to retrieve at a particular location. For example, in Figure 4a, the user can activate Jarvis using a wake word: ***"Hi Jarvis"***. Next, Jarvis answers by saying in speech and text "***Yes!***". Then, the user can ask Jarvis by saying ***"What is the temperature in the living room?"***. The speech recognition component of Jarvis takes the voice question and converts it into a text transcription, which is in turn converted by the Text-to-CDQL component to its corresponding CDQL query ***"pull (device.stateValue) define entity device is from smart home where device.hasState.isOn = true and device.deviceName = "temperature" and device.isIn.buildingenvironment = "living room")"***. Subsequently, the CDQL query is sent to the platform CooaS via a POST request to extract the value of the temperature in the living room, and then, CooaS returns the data to the UI. Jarvis will show the answer in the UI by mentioning ***"The temperature in the living room is 20 degrees celsius"***. Apart from asking the temperate of one specific room, the user can ask about the volume level of the smart TV as shown in Figure 4a.

Also, as shown in the scenarios of Figures 4b and 4c, the user can ask about the state of one smart device by saying ***"Is the living room light on?"*** or ***"Give me the state of the light in the living room?"***. Jarvis may answer ***"Yes, the light in the living room is turned on."***.

B. Complex scenario

In addition, we also plan to demonstrate a more complex scenario that allows a user to ask Jarvis more contextual questions. For example, the user can ask ***"What is the current temperature?"***. However, Jarvis cannot easily understand for what room the user wants to know the temperature. Therefore, Jarvis will ask the user clarification questions ***"Where do you want to know the temperature?"***. Then, Jarvis will act like the base scenario if the user answers with a specific location of the building environments, such as the living room. On the other hand, if the user answers ***"Give me please the average temperature"***, Jarvis will extract all data from the temperature sensors in the smart home and takes the average number as the final answer that will be presented to the user as in Figure 5.

IV. Conclusion and Future Work

This paper introduces Jarvis, a context-as-a-service mobile tool with a voice-based UI, which enables context-aware data collection, service discovery, and computer-aided situational awareness through a conversational User Interface (UI). Jarvis relies on two main components: a conversational user interface to converse, and an operational component to enable smart things and IoT silos to discover, validate and share relevant and dependable context. Future work consists of improving Jarvis to handle more complex natural language queries and to improve the performance of the system. We also plan to extend the applicability of the tool to other applications such as smart city, business scenarios, or missions-critical applications [13].

Fig. 5: Complex scenario.

References

[1] A. Nordrum *et al.*, "Popular internet of things forecast of 50 billion devices by 2020 is outdated," *IEEE spectrum*, vol. 18, no. 3, 2016.

[2] N. Dey, A. E. Hassanien, C. Bhatt, A. Ashour, and S. C. Satapathy, *Internet of things and big data analytics toward next-generation intelligence*. Springer, 2018, vol. 35.

[3] L. Jiang, D.-Y. Liu, and B. Yang, "Smart home research," in *Proceedings of 2004 international conference on machine learning and cybernetics (IEEE Cat. No. 04EX826)*, vol. 2. IEEE, 2004, pp. 659–663.

[4] M. Alaa, A. Zaidan, B. Zaidan, M. Talal, and M. Kiah, "A review of smart home applications based on internet of things," *Journal of Network and Computer Applications*, pp. 48–65, 2017.

[5] S. Sagiroglu and D. Sinanc, "Big data: A review," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 42–47.

[6] A. Hassani, A. Medvedev, P.-D. Haghighi, S. Ling, A. Zaslavsky, and P.-P. Jayaraman, "Context definition and query language: Conceptual specification, implementation, and evaluation," *Selected Papers from the 2nd Global IoT Summit: IoT Technologies and Applications for the Benefit of Society)*, 2019.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[11] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015.

[12] M. Post and D. Vilar, "Fast lexically constrained decoding with dynamic beam allocation for neural machine translation," 2018.

[13] N. D. Huynh, M. R. Bouadjenek, I. Razzak, K. Lee, C. Arora, A. Hassani, and A. Zaslavsky, "Adversarial attacks on speech recognition systems for mission-critical applications: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2202.10594