Intrusive plug management system using chatbots in office environments

Ramasubbu, Dhineshkumar; Baskaran, Krishnamoorthy; Yann, Grynberg

2018

Ramasubbu, D., Baskaran, K., & Yann, G. (2018). Intrusive plug management system using chatbots in office environments. Proceedings of the 2018 Asian Conference on Energy, Power and Transportation Electrification (ACEPT), 1-4. doi:10.1109/accept.2018.8610869

<https://hdl.handle.net/10356/143994>

<https://doi.org/10.1109/ACEPT.2018.8610869>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: <https://doi.org/10.1109/ACEPT.2018.8610869>.

Downloaded on 26 Apr 2023 19:40:27 SGT

Intrusive Plug Management System Using Chatbots in Office Environments

Dhineshkumar Ramasubbu
Vellore Institute of Technology
Chennai Campus,
India
r.dhineshkumar2014@vit.ac.in

KrishnamoorthyBaskaran, Grynberg Yann
Energy Research Institute @ NTU (ERI@N)
Nanyang Technological University,
Singapore
kbaskaran@ntu.edu.sg, ygrynberg@ntu.edu.sg

Abstract— “Only 44 percent of computers, 32 percent of monitors, and 25 percent of printers were turned off at night” [1], with energy efficient appliances employed in office environments, occupant’s energy-conscious behavior plays a vital role in monitoring the plug load. In an attempt to involve the occupants to the building’s energy management suite, a natural language-based plug management system is proposed. This article aims to develop a rule-based chatbot that helps users manage (schedule) their plugged-in appliances through smart plugs in an office environment. Considering the nature of the application and the accuracy of the intended operation, a rule-based chatbot is developed to schedule the smart plugs. It is developed using Python to be integrated with instant messaging application Slack.

Keywords— *NLP, Smart plug, Plug load management system, Chatbot.*

I. INTRODUCTION

Among the various energy consumers, it is reported that in 2017 over 39% of total U.S energy consumption was consumed by the residential and commercial sector in the USA [2]. There is a rise in need of cyber-physical energy management systems (EMSs) for modern commercial buildings composing both micro-grid and Internet of Things (IoT) framework. With the primary goal to conserve electrical energy, many EMS systems have been proposed [3][4] and few of them have been implemented in commercial spaces. While the traditional EMSs efforts to conserve energy were concentrated mostly on the broader spectrum like Heating, Ventilating, Air Conditioning (HVAC) systems and hard-wired lightings, energy impact of plugged in appliances in the office spaces has received attention only recently and studied in various aspects [5][6].

Electrical plug-load monitoring of appliances in commercial buildings can mitigate the energy impact of plugged in device’s contribution, over 23% [7], to the overall building’s electrical energy consumption. With increased energy efficient appliances deployed in office spaces, one of the viable ways to address the issue is through the occupant’s energy-conscious behavior to control and monitor their appliances more efficiently [8]. Usually, the plug loads are monitored either in a non-intrusive or intrusive way [5]. Intrusive Load Monitoring (ILM) methods involve installing individual meters typically smart plugs at desired plug points to measure the energy consumption from the appliances [9]. Contrary to the non-intrusive methods [10][11], ILM offers appliance-level monitoring of energy consumption leading to an informed decision-making process [9]. Interfacing

the users to the system plays an important role in executing effective ILM. Typically, the web or mobile based applications are required to make an occupant interact with his/her devices.

Natural Language Processing (NLP), an important subfield of Artificial Intelligence (AI) that deals with natural language information, has been extensively used in practical applications that involve text classification [12], document summarization [13], language translation [14] etc. Similarly, they also play an important role in Human-Computer Interaction (HCI)[15]. A chatbot is one such example and it has been an attractive application of NLP to the enterprise and commercial sectors [16]. They play a very vital role in internal and external organization management. It is widely used to handle customer inquiries, collect user information, arrange for meetings and in some cases, it is an interactive alternative to web-based system interaction. With the substantial use of instant messaging applications such as Microsoft Teams, Slack, Flock etc., by the corporate firms, especially for knowledge sharing among the teams, the fullest potential of chatbots is liberated.

This work aims to explore a different use case of chatbots in various office environments to manage plug load from an end user perspective and eventually conserve energy. The rest of the article is organized as follows. The overview of the proposed system and the steps involved in the system development is described in section 2 and the experimental details of the system are explained in section 3. Section 4 describes the conclusion and future enhancements to the system.

II. PROPOSED SYSTEM

This work addresses a way to integrate AI based chatbot to provide a text-based plug management system for users in an office environment. In this article, the design and development of a rule-based chatbot system for plug scheduling are discussed. Fig. 1 shows the overview of the proposed system.

The system consists of three major entities which interact with each other to perform the scheduling task intended by the user. The entities include 1. Chatbot, 2. Database, and 3. Scheduler. The Chatbot receives the text input from the user to extract necessary scheduling information and record it in the Database. On the other hand, the Scheduler periodically reads the Database to actuate the scheduling task intended by the user.

The chatbot designed for this application is a simple rule-based program that receives a formal text input from the user and stores the scheduling information from the input in the

Fig. 1. System Overview.

Database. Considering the chatbot's only intended function to schedule user's plugs and the office environment in which the system is being deployed, a rule-based approach is chosen for development. The text input received from the user is first matched against a set of specified rules also known as grammar to extract the required information such as plug identity, time to schedule and an action (on/off) for scheduling.

When dealing with a grammar-based approach for natural language understanding, a text input that adheres strictly to the specified grammar is expected. Though the conversation between the user and the bot might not be as natural as a conversation between two human users, the accuracy of achieving the intended task is expected to be quite satisfactory [17]. Fig. 2 shows the flow diagram of the steps involved in the development of chatbot.

The primary step is to pre-process the input text. In this step, the text information received from the user is filtered in such a way that in the further steps only the information required for the scheduling task is processed. This includes converting all uppercase characters to lowercase characters, discarding irrelevant or unnecessary special characters (?, !, <, >, etc.), and trimming off duplicate or trailing white spaces, periods, and commas. The most important part of pre-processing the input text is tokenization. The input string is split into a meaningful sequence of characters called tokens which are further processed to extract context and application-specific information. For languages like English, the input text is usually segmented into words (tokens) and processed further.

In the next phase, Syntax Analysis, the pre-processed input text is matched against a predefined grammar to verify whether the input is grammatically structured for the scheduling task. This process is called parsing. The grammar consists of production rules to describe all the valid tokens to match the input text with. If the input text can be derived with the production rules of the grammar beginning with the start symbol, then the input is said to be a grammatically valid input. Knowledge about the potential candidates for the input text can play a vital role in designing a well-defined grammar. As mentioned earlier, the important information looked out for in the input text are the plug identity, specific action (on/off), and duration. Fig. 3 shows the sample inputs that can be expected from a user to perform the scheduling task. These phrases are considered specifically because the information required for the task is mentioned in a concise, natural and unambiguous way possible.

As can be seen from Fig. 3, not all phrases have all the specific information required for a scheduling (statement 1 and statement 3) and it is possible that the user might not have all the required information, especially when a user has control over more than two plugs. To help complete the task initiated by the user in such situations, even the incomplete phrases are considered while constructing the grammar. The missing information is then gathered by questioning the user leading to a chat thread. In addition, the user might want to schedule different plugs at the different duration and similarly different plugs at the same time. To accommodate such possibilities, complex sentence structures incorporating conjunction words are also considered for grammar construction.

To validate an input text from the user that has a similar sentence structure to the phrases described in Fig. 3, a Context Free Grammar (CFG) is designed specifically. Fig. 4 shows an extract of the constructed CFG that produces statement 2 of Fig. 3 when parsed. The production rules are fired from the start symbol S in a sequence. Fig. 4 also shows the parse tree generated for the statement 2 when parsed with the CFG. It can be noted that the grammar constructed is unambiguous; for a given input text there can be only one parse tree or parse sequence possible with respect to the grammar. Designing an unambiguous grammar is crucial because the way in which the

Fig. 2. Flow diagram of the steps involved in chatbot development.

1. Switch off my plugs
2. Switch on Plug1 from 3.00 AM to 5.00 PM
3. Switch on p1 and p2
4. Switch off p1 from 1:30 PM to 2:00 PM
5. Switch on p1, p2 and p5 from 10:00 AM to 2:00 PM
6. Switch off p1 from 16:20 to 18:00 and switch on p2 from 8:00 PM to 6:00 AM

Fig. 3. Sample input text messages expected from the user.

system interprets the input information should be crisp and unambiguous considering substantial ambiguity of the English language itself.

In the next phase, the logical interpretation of the input text including the necessary scheduling information is extracted. This step is usually carried out simultaneously in the syntax analysis phase by augmenting additional attributes to the grammar to capture the logical information and features from the text. Fig. 5 shows the attribute grammar for the grammar described in Fig. 4. The role of the semantic analysis phase is quantified when dealing with complex sentence structures such as statement 5 and 6 in Fig. 3. In such cases, preserving the contextual information of the individual subjects in the input text is important for an unambiguous interpretation. The extracted features are collated to be stored in a database. This is then retrieved by the Scheduler for necessary actuation in the smart plugs.

Fig.4. An extract of the grammar and a sample parse tree.

III. EXPERIMENTAL SPECIFICATION

The proposed system is designed specifically for an office environment where the employees have control over at least one smart plug and the devices that are connected to the smart plugs typically include computer monitors, CPUs, and laptops. The TPLINK HS110 smart plugs were used for the experiment and they are remotely controlled through the open-source 'tplink-smarthome-api' [18] API. In total, 40 smart plugs were installed in an office environment and connected to a local network. The programming language used for development is Python 3.5 and the chatbot interface is designed and developed to be operated in Slack. Slack is a software application that helps members of a group or team to communicate and collaborate through a common channel. The chatbot and the users handling their smart plugs share the same Slack workspace. The user provides instruction to the chatbot by referencing it using '@' followed by the name of the chatbot in the text message.

A web server is set up using Flask API [19] to interface Slack with the scheduling logic. The Slack Events API is used to capture the message events in the workspace for the chatbot to react promptly. Only message events that have an explicit reference to the chatbot is considered for further processing. The text message is then processed to complete the task intended by the user. The Natural Language Toolkit (NLTK) [20] is used to process the received text; from defining an attribute augmented grammar to extracting features from the

text. When dealing with text inputs which lack complete information required for the scheduling, the missing details are then inquired from the user using Dialogs. It is a menu-based user interface in Slack to prompt input from the user, this is made personalized to individual user based on their profile which includes user identity and the number of plugs they have control over. The data provided by the user in the Dialog interface is sent directly to the server and the server decides the action to be performed on the data. Once all the required information for the scheduling is received, the server updates the database with a new record which is then retrieved by the scheduler to instruct the smart plugs over Wi-Fi.

S -> X P	[S.schedule_details=(X.action_details,P.plug_details)]
X -> 'switch' A	[X.action_details= A.action]
A -> 'on' 'off'	[A.action='on']
A -> 'off'	[A.action='off']
P -> PN D	[P.plug_details=(PN.plug_name,D.time_details)]
D -> F K	[D.duration=(F.from_time,K.to_time)]
PN -> 'Plug1'	[PN.plug_name='Plug1']
F -> 'from' T	[F.from_time= T.time]
K -> 'to' T	[K.to_time= T.time]
T -> time_format	[T.time= time(time_format)]
time_format=	^(([0-9]{1-9} [0-2])(:\.)([0-5][0-9]()?(am pm))\$)

Fig.5. Attribute augmented grammar for semantic analysis and feature extraction.

IV. CONCLUSION & DISCUSSION

With energy efficient appliances employed in green office environments, the occupant's energy-conscious behavior plays a vital role in monitoring the plug load which contributes over 23% of total electrical energy in commercial buildings. Involving the occupants directly to the building's EMS will help cultivate such behavior. A chatbot based plug scheduling system is considered in this article; the user has direct control over his/her appliances which are connected to the smart plugs in an office environment through a text-based interface. The system is designed and developed to be integrated to instant messaging applications which the occupants use on day-to-day basis. This engages the occupants in the building's EMS and enables them to manage their office appliances in the most natural and efficient way possible eventually leading to conserving energy.

Future enhancements to the system include employing a machine learning based text processing. Contrary to the rule-based processing, the level to which the ambiguity of the language handled through machine learning especially with the deep neural network is far better than a rule-based chatbot. The system can be scaled further to look for text messages in a thread that has indirect information about a possible scheduling task. For example, there might be a situation where a user informs the team members that he/she could not make it to the office tomorrow for a specific reason through a text message and the system processing such texts should prompt the user to schedule their smart plugs on that specific day automatically. In this way the user saves time and the organization saves electrical energy.

ACKNOWLEDGEMENT

This work is in part of the “Intelligent plug-load management system for smart office environment” project carried out at Energy Research Institute @ NTU, Nanyang Technological University, Singapore.

REFERENCES

- [1] C. A. Webber, J. A. Roberson, R. E. Brown, C. T. Payne, B. Nordman, and J. G. Koomey, “Field surveys of office equipment operating patterns; TOPICAL”, Ernest Orlando Lawrence Berkeley National Laboratory, 2001.
- [2] Monthly Energy Review July 2018, Energy Consumption by Sector- U.S. Energy Information Administration.[Online].
Available: <https://www.eia.gov/totalenergy/data/monthly/pdf/sec2.pdf>
- [3] S. Katipamula, R. M. Underhill, J. K. Goddard, D. J. Taasevigen, M. A. Piette, J. Granderson, R. E. Brown, S. M. Lanzisera, and T. Kuruganti, “Small-and medium-sized commercial building monitoring and controls needs: A scoping study”, Pacific Northwest National Lab, Richland, WA (United States), 2012.
- [4] F. Behrooz, A. R. Ramli, and Khairulmizam Samsudin, “A survey on applying different control methods approach in building automation systems to obtain more energy efficiency”, *International Journal of Physical Sciences*, vol. 6, no. 9, pp. 2308-2314, 2011.
- [5] A. Ghosh, K. A. Patil, and S. K. Vuppala, “PLEMS: Plug load energy management solution for enterprises”, in *Advanced Information Networking and Applications (AINA)*, 2013 IEEE 27th International Conference, 2013, pp. 25-32.
- [6] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. MP. O'Hare, “Real-time recognition and profiling of appliances through a single electricity sensor”, in *Sensor Mesh and Ad Hoc Communications and Networks (SECON)*, 2010 7th Annual IEEE Communications Society Conference, 2010, pp. 1-9.
- [7] California Commercial End-use Survey: Appendices AJ. vol. 2, 2006.[Online].
Available: <https://www.energy.ca.gov/2006publications/CEC-400-2006-005/CEC-400-2006-005.PDF>
- [8] J. Zhao, B. Lasternas, R. Yun, C. Zhang, H. Wang, A. Aziz, K. Lam, and V. Loftness, “Behavior oriented metrics for plug load energy savings in office environment”, in *Proceedings of the 2014 ACEEE Summer study on Energy Efficient Buildings*, 2014.
- [9] A. Ridi, C. Gisler, and J. Hennebert, “A survey on intrusive load monitoring for appliance recognition,” in *Pattern Recognition (ICPR)*, 2014 22nd International Conference, pp. 3702-3707, 2014.
- [10] G. W. Hart, “Nonintrusive appliance load monitoring”, in *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870-1891, 1992.
- [11] J. R. Herrero, Á. L. Murciego, A. L. Barriuso, D. H. Iglesia, G. V. González, J. M. C. Rodríguez, and R. Carreira, “Non Intrusive Load Monitoring (NILM): A State of the Art”, in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 2017, pp. 125-138.
- [12] J. C. Na, H. Sui, C. S. G. Khoo, S. Chan, Y. Zhou, “Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews”, in *Proceedings of the Eighth International ISKO conference*, 2004.
- [13] M. Hu, A. Sun, and E. P. Lim, “Comments-oriented document summarization: understanding documents with readers' feedback”, in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 2008, pp. 291-298.
- [14] G. Lemnbersky, N. Ordan, and S. Wintner, “Language models for machine translation: Original vs. translated texts”, in *Computational Linguistics*, vol. 38, no. 4, pp. 799-825, 2012.
- [15] B. Manaris, “Natural language processing: A human-computer interaction perspective”, in *Advances in Computers*, vol. 47, pp. 1-66, 1998.
- [16] H. S. Nwana, “Software agents: An overview”, *The knowledge engineering review*, vol. 11, no. 3, pp. 205-244, 1996.
- [17] H. N. Io, and C. B. Lee, “Chatbots and conversational agents: A bibliometric analysis”, in *Industrial Engineering and Engineering Management (IEEM)*, 2017 IEEE International Conference, 2007, pp. 215-219.
- [18] TP-Link Smarthome WiFi API.[Online].
Available: <https://github.com/plasticrake/tplink-smarthome-api>
- [19] Flask API.[Online].
Available: <https://www.flaskapi.org/>
- [20] E. Loper, and S. Bird, “NLTK: The natural language toolkit”, in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics*, vol. 1, pp. 63-70. 2002.