Applying an Intelligent Personal Agent on a Smart Home Using a Novel Dialogue Generator

Anastasios Alexiadis(✉), Alexandros Nizamis, Ioannis Koskinas, Dimosthenis Ioannidis, Konstantinos Votis, and Dimitrios Tzovaras

Centre for Research and Technology Hellas, Information Technologies Institute (CERTH/ITI), 6km Charilaou-Thermi, Thessaloniki, Greece
{talex,alnizami,jkosk,djoannid,kvotis,Dimitrios.Tzovaras}@iti.gr

Abstract. Nowadays, Intelligent Personal Agents include Natural Language Understanding (NLU) modules, that utilize Machine Learning (ML), which can be included in different kind of applications in order to enable the translation of users' input into different kinds of actions, as well as ML modules that handle dialogue. This translation is attained by the matching of a user's sentence with an intent contained in an Agent. This paper introduces the first generation of the CERTH Intelligent Personal Agent (CIPA) which is based on the RASA (https://rasa.com/) framework and utilizes two machine learning models for NLU and dialogue flow classification. Besides the architecture of CIPA—Generation A, a novel dialogue-story generator that is based on the idea of adjacency pairs is introduced. By utilizing on this novel-generator, the agent is able to create all the possible dialog trees in order to handle conversations without training on existing data in contrast with the majority of the current alternative solutions. CIPA supports multiple intents and it is capable of classifying complex sentences consisting of two user's intents into two automatic operations from the part of the agent. The introduced CIPA—Generation A has been deployed and tested in a real-world scenario at Centre's of Research & Technology Hellas (CERTH) nZEB Smart Home (https://smarthome.iti.gr/) in two different domains, energy and health domain.

Keywords: Intelligent agents · Natural Language Understanding · Multiple intents · Smart home

1 Introduction

Fast growth in natural language understanding capabilities of intelligent virtual agents enables its wide use on different types of consumers devices. This type of personal agents have been the focus of research for a long time with plenty of significant results. ELIZA [1] and IBM Shoebox [2] are considered as two of the most representative research's first outcomes. Nowadays, there is a wide variety of Intelligent Personal Assistants (IPAs) such as Google Assistant, Amazon

© IFIP International Federation for Information Processing 2020
Published by Springer Nature Switzerland AG 2020
I. Maglogiannis et al. (Eds.): AIAI 2020, IFIP AICT 584, pp. 384–395, 2020.
https://doi.org/10.1007/978-3-030-49186-4_32

Alexa and Apple Siri that attempt to assist humans either by providing them with information or by performing specific tasks for them as a results of their communication with humans using natural language. These systems aim to solve two different but related problems: (a) the natural language understanding and (b) the management of the flow of the conversation (dialogue). Based on these two aforementioned problems, different frameworks have been designed. These frameworks provide to researchers and developers systems that are capable of handling the above problems in order to enable them to develop Virtual Assistants and chat-bots for specific domains. These IPAs frameworks include Google Dialogflow, Amazon Lex, Facebook WIT.AI and RASA. Most of these frameworks are accessed through web Application Programming Interfaces (APIs) and the developers do not have access neither to their source code or to the machine learning models that power them.

Leveraging on the above described research results and frameworks, the first version of CERTH Intelligent Personal Agent is designed and introduced in this work. The main key innovative aspect of the proposed agent is the adoption of a novel dialogue-story generator that is based on the idea of adjacency pairs and is able to cover all the possible states in the communication between the humans and the agent. Furthermore, CIPA Generation A is able to support applications of different domains and it is available in a multi-intent model.

The paper is structured as follows. Following the Introduction, a brief literature review is presented. The proposed CIPA–gen A's architecture is analyzed in Sect. 3. In the same section, the dialogue-story generator that is based on the idea of adjacency pairs is presented as well. Section 4 consists the evaluation section and provides a brief description of the use cases of both domains, energy and health. Finally, the conclusions and future work are drawn in Sect. 5.

2 Related Works

Modern conversational agents relies on natural language processing, supported by AI and ML techniques and execute tasks based on verbal interaction with end-users. As the proposed work is driven by results of research and development in the fields of task-based agents, conversational agents, and dialogue generation, a brief analysis of related works in these fields is documented in this section.

The proposed work is strongly related with dialogue generation in general. Modeling the future direction of a dialogue is crucial to generate coherent dialogues and execute tasks. To this aim, data-driven conversation modeling mostly build upon recurrent neural networks (RNNs) [13] that enable an agent to learn how to map rules between input messages and responses from a massive amount of training data. StarSpace [3] is a general-purpose neural embedding model that can solve a wide variety of problems including text classification. The StarSpace model consists of learning entities that are described by a set of discrete features. Its main contributions are (a) the generalization across diverse problems and (b) the comparison between embeddings of different types. Industry leaders such as Google Assistant, Amazon Alexa and Apple Siri are based on ML models for

dialogue generation. Recently, Google introduces Meena[1], an end-to-end, neural conversational model that learns to respond sensibly to a given conversational context. Meena has a single evolved transformer encoder block for processing the conversation context block and 13 evolved transformer decoder blocks to formulate an actual response. The Encoder-Decoder Model for dialogue generation is a popular technique that is adopted from research community as well. Serban et al. [11] introduces an encoder-decoder model for generating dialogues. It is based in a neural network-based architecture, with hierarchical latent variables that capture dependencies over an extended conversation history. To the same direction, a variational hierarchical conversation RNNs model is introduced in [12]. Other recent approaches are based on Reinforcement Learning techniques. Li et al. [14] train two models, a generative model to produce response sequences, and a discriminator to distinguish between the human-generated dialogues and the machine-generated ones. The generative model receives the output of the discriminator as a reward in order to push the complete system to generate dialogues that mostly resemble human dialogues. In [15] deep reinforcement learning for dialogue generation method was lately proposed. The method optimizes long-term rewards and uses a encoder-decoder architecture. It simulates the conversation between two virtual agents to discover possible actions while learns to maximize expected reward. Furthermore, plenty approaches related to task-oriented RL methods have been proposed [16].

Besides the general works related to dialogue generation, smart home automation applications that were led by task oriented dialogue systems have been introduced. Recently, Park et al. [17], introduced a framework for development of task-oriented dialogue systems in a Smart Home environment. Domain knowledge is required for the proposed task-oriented dialogue system. The framework ontologically expresses the required knowledge and is able to build a dialogue system by editing the dialogue knowledge. In addition to this, a more intelligent conversation is enabled by providing a hierarchical argument structure to manage the various argument representations that are included in natural language sentences. The dialogue management system of the introduced framework is based on a rule-based systems. Rule-based dialogue management systems are defining the possible states of a dialogue and the behavior of the system for the given state. Both finite-state [8] and information-state [9] based systems are used. Especially, the finite-state methodology is highly-correlated to the adjacency pairs methodology, that is introduced from the presented work, as a concept for dialogue generation. However both finite-state and information-state techniques are vulnerable to errors in NLU, because they are entirely dependent on the result of the NLU to recognize the intent of the user. A smart home automation system, named Cassandra, has been introduced in [10]. A voice assisting system based on automatic speech recognition and NLU is proposed in order to enable user to control smart home appliances. Opposite to the presented work in this paper, Cassandra uses a knowledge base of predefined scenarios to handle the dialog with the user. After analysis and parameters' extraction based on user's

[1] https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html.

voice command, a predifined scenario is selected. Furthermore, the end-user is able to alter Cassandra's configuration by creating scenarios or editing existing ones.

In contrast with the above described related works, that are based on dialogue generation for existing knowledge and data, the proposed CIPA Gen—A introduces a novel dialogue generator that does not require the availability of historical data. It is based on adjacency pairs and enables the generation of all the possible dialogue trees. Adjacency pairs were first introduced by Schegloff and Sacks [18] as the basic foundation of conversational structure can be used for the evaluation of dialogue stories. Midgley et al. [7] proposed a new method of dialogue segmentation in order to leverage the gathered information from more than one previous utterances identifying adjacency pair labels that occur with high frequency.

3 CIPA Gen—A Architecture

In this section the design approaches of CERTH Intelligent Personal Agent Gen–A architecture are presented. CIPA–gen A is designed in order to support multiple task domains. This capability is enabled by auto-generated Python scripts that sub-class RASA Core's Action classes, one class for each action included in a CIPA domain, and by providing a call-back function that handles the actions for that domain. Moreover, CIPA–gen A offers an API that supports exchange of messages in JSON format. Furthermore, it includes an SQL database for the storage of user's messages and an action history. CIPA–gen A is provided with a multiple intent model. The novel dialogue generator from CIPA–gen A and the supported multi intent model are explained in details in the following subsections. The technical descriptions come alongside with examples related to the supported domains by the Agent and their datasets in order to describe better the application of the proposed agent and be easier for the reader to understand and follow the concepts in this document.

3.1 Supported Domains and Datasets

CIPA–gen A has been deployed in CERTH nZEB Smart Home in two different domains, energy and health. The CERTH nZEB Smart Home is a rapid prototyping and a novel technologies demonstration infrastructure resembling a real domestic building where occupants can experience actual living scenarios while exploring various innovating smart IoT-based technologies with provided Energy, Health, Big Data, Robotics and Artificial Intelligence (AI) services.

CERTH nZEB Smart Home - Energy Domain: Smart Home is equipped with energy domain related IoT devices that monitor the energy consumption and production, and the conditions of the entire building while various algorithms can support automation and energy efficiency scenarios. A dataset has been created from the testers' (Smart Home occupants) inputs, thereafter referred to as the *gold* set of the energy domain. This set is continuously extended

for nZEB Smart Home's energy domain. Due to its size (1608 samples) it is solely used as a test set. A common pattern that has been found in the data was the omission of information crucial for an operation to be performed. For example a user could utter "Turn on the lights" without providing the room where he/she wanted the action to be performed. Moreover, there were observations such as "Turn the lights" which did not contain the fully specified action information, such as if the user wanted the lights on or off. As the user could request a change for the state of the lights of another room this is crucial information.

CERTH nZEB Smart Home - Health Domain: Health related IoT devices monitors a variety of physiological attributes, and enabling the extraction of valuable data through intelligent processing towards preventing situations that could lead to harmful outcomes. A dataset, based on the testers' inputs, has been created. Thereafter, it is referred as the *gold* set of the health domain. This set is continuously extended and updated for nZEB Smart Home's health domain. Due to its size (1972 samples) it is solely used as a test set.

3.2 Novel Dialogue Generator

A novel dialogue-story generator that is based on the idea of adjacency pairs has been designed and developed for CIPA–gen A. The proposed story generator models dialogues that consist of a subset of adjacency pairs based on the following two assumptions: (a) the user may omit information required by an action and thus the Agent have to ask for that information by interacting with the user in a dialogue and (b) the user may not cooperate by following up the conversation. For example, instead of replying to a question posed by the Agent, the user may request another action.

For every action requested, the story generator produces conversation flows with various pieces of the missing but required information and defines the conversation flows that interact with the user to obtain this missing information. This produces all valid conversation flows for each action. For example, consider the "Turn on hvac" action from the nZEB Smart Home energy domain. The introduced generator will produce the following valid stories for this action:

- The user gives the room and the on/off switch.
- The user gives only the room and the agent asks for the on/off operation.
- The user gives only the on/off operation and the agent asks for the room.
- The user doesn't give the required information and the agent first asks for the room and after getting a correct reply asks for the on/off option.

In addition, the generator produces dialogues for handling valid multi-intents by the user. For example in the `turn-hvac+change-hvac-mode` multi intent, the generator will produce all valid adjacency pair flows for the union of their slots. At the end of each story it will call each action sequentially. If a slot is common to both intents and the user has not provided it, the CIPA–gen A will ask it once.

Algorithm 1 Write Story

```
1: procedure WRITE STORY(intent, intenToProcess, included,
   excluded, runRepeatedAct, mappings, c, repeat)
2:     s ← 0
3:     if repeat > −1 then
4:         write repeat signature number c
5:     elsemultiIntent(intent)
6:         write multi intent signature number c
7:         write single intent signature number c
8:     end if
9:     WriteStoryII(intent, included, mappings)
10:    for i ← 0 to excluded.length do
11:        write ' -' slots_to_act_map[excluded[i]]
12:        if repeat = i and runRepeatedAct = False and s = 0 then
13:            WriteStoryII(intentToProcess, [ ], mappings)
14:            write ' - utter_repeat'
15:            break
16:        elserepeat = i and s = 0
17:            WriteStoryII(intentToProcess, [excluded[i]], mappings)
18:            s ← 1
19:            continue
20:        end if
21:        write '* inform{' (excluded[i], slot_example_fun(excluded[i])) '}'
22:    end for
23:    if !multiIntent(intent) and (repeat = −1 or runRepeatedAct = True) then
24:        write ' -' intent_to_act_map[intent]
25:    elserepeat = −1 or runRepeatedAct = True
26:        for each intent inte_ of multi intent intent do
27:            write ' -' intent_to_act_map[inte_]
28:        end for
29:    end if
30:    write ' - action_restarted'
31: end procedure
```

Moreover, it produces invalid conversation flows in which the user does not cooperate by following up the conversation with the supported intents but he/she follows up the conversation with a different intent. The invalid conversation flows restart the conversation after the agent tells the user that it did not understand his/her intentions.

The proposed algorithm that generates stories (Algorithm 1 to 3), takes as inputs an intent to action mapping, a function that maps intents to a set of slots (that are needed for the intent), a slot example relation that maps slots to random slot instances, a slot to action mapping (that specifies which action requests each slot) and a mapping of valid multi intents. The algorithm automatically generates the story file (in the RASA format) for training the dialogue models. It operates by including a tree-expansion phase to produce all possible story combinations that have slot information missing in the original user message, as well as all invalid variations of them in which the user's reply is classified to a different intent from the expected one.

Algorithm 2 Generate all stories of an intent

```
1: procedure GEN INTENT(intentToProcess, rc, mappings)
2:     if multIntent(intentToProcess) then
3:         Split intentToProcess to individual intents
4:         slots ← union of slots of the individual intents
5:     else
6:         slots ← slots of intentToProcess
7:     end if
8:     c = 1
9:     slotsComb = SlotsCombinations(slots)
10:    for include, exluded in slotsComb do
11:        WriteStory(intentToProcess, intentToProcess, included,
   excluded, runRepeated = False, mappings, c, −1)
12:        c ← c + 1
13:        for excludedItem[i] in excluded do
14:            for intent in intent_to_act_map do
15:                if intent ≠ intentToProcess then
16:                    WriteStory(intentToProcess, intent, included,
   excluded, runRepeatedAct = False, mappings, rc, i)
17:                else
18:                    WriteStory(intenToProcess, intent, included,
   excluded, runRepeatedAct = True, mappings, rc, i)
19:                end if
20:                rc ← rc + 1
21:            end for
22:        end for
23:    end for
       return rc
24: end procedure
```

3.3 Single and Multi Intent Models

Both single and multi intent models are supported by the CIPA–gen A. In the single intent model the Chatito, a third-party open-source natural language generator is used in order to create a training set for the NLU module. For the NLU model we did not select Rasa NLU's default model that uses the SpaCy Natural Language Processing system but opted to use a RASA NLU Model based on StarSpace for intent classification. Afterwards, we defined in Chatito's DSL (Domain Specific Language) examples of intents for the nZEB Smart Home domain. These do not correspond to an exact one to one mapping, as we defined some extra intents. The two most important of them are the *inform* and *nounderstand* intents. The first should match all user messages that inform the agent about a slot (a parameter) of an action, such as the location of an action, in the case it was not included in the original message. The second one, the nounderstand intent, should match all user messages that resemble messages that correspond to other intents but that do not make sense. For example consider the user message "turn on the lights" and the message "turn on the door", the first message should be classified to the *turn the lights* intent, whereas the second

should be classified to the *nounderstand* intent. By specifying these intents and text patterns in the Chatito DSL, we used the generator to generate our training set for the NLU Model.

For the dialogue Model, we used the default Rasa Core's LSTM and memorization models. Furthermore, we added a fallback policy for both the intent classification and dialogue classification models. Therefore, the Agent will be able to reply that he does not understand anything in the cases of low classification confidence.

Algorithm 3 Story Gen - Main function of the algorithm

```
1: procedure STORY GEN(domain_mappings)
2:     Write a story of no understand classified to utter default
3:     rc ← 1
4:     for intent in intent_to_action_map do
5:         rc ← GenIntent(intent, rc, domain_mappings)
6:     end for
7:     for (inte1 → intents) in multi_intent_map do
8:         for inte2 in intents do
9:             rc ← GenIntent(inte1′ +′ inte2, rc, domain_mappings)
10:        end for
11:    end for
12:    for slot in slots_to_act_map do
13:        Write a story consisting of a single inform and a slot
       classified to no-understand
14:    end for
15:    Write a story consisting of a single inform classified to no-understand
16: end procedure
```

For the multi intent model we developed an algorithm that takes as input a Chatito DSL file alongside with a mapping of valid multi intents and then, it produces an extended Chatito DSL file that includes patterns of these multi intents by combining the patterns of their single intents counterparts. Moreover, it uses a random variable to consider omitted duplicated words when combining messages from two intents. For example, in the case of combining "please turn on the hvac in the kitchen" with "turn the hvac mode on", the second "hvac" word will probably be omitted in the training example for the multi-itent.

Thereafter, we used the output of this algorithm to generate our NLU training examples using Chatito. The output of this algorithm produced a multitude of multi intent data that surpassed the single-intent ones in the data. To balance the classes we post-processed this output by tripling to quadrupling the single-intent observations in the data.

Furthermore, we extended RASA NLU's adaption of the StarSpace method, by adding more hidden layers to the NLU model and training an NLU Model for multi-intent classification. We used a $512 \times 384 \times 256 \times 128$ configuration for network A and a 384×128 configuration for network B.

4 Evaluation

In this section the evaluation results for the multi intent models and for both domains, energy and health, are presented. Prior to introducing the results of the

evaluation, an assumption for miss-classification should be taken into consideration. Most of the miss-classification were of the type `no-understand` to `inform`. This miss-classification type is not a problem as it is handled by the dialogue model. We can classify orphan `inform`s (that is `inform`s that do not occur in the middle of the dialogue) as `no-understand`s, whereas incorrect `inform`s in the middle of a dialogue that do no match any of the valid dialogue flows generated by the generator presented in this paper are handled by the produced stories for invalid scenarios. So, if we consider this type of miss-classifications as correct our accuracy is increased (column *Accuracy2* in Table 1).

Table 1. Evaluation metrics

Evaluation	Accuracy	Accuracy 2	Lowest conf
M-Intent (Energy)	98.39%	99%	0.53
M-Intent (Health)	99.14%	99.75%	0.47

Table 1 presents the results of the multi-intent NLU model on the gold sets of the two domains. We have included various text patterns for these intents in the Chatito DSL and generated our training data for the NLU Models as we did not have access to enough real world user messages for these domains. Training our NLU Models on real-world data should provide a stronger model for these domains.

To evaluate the dialogue generator we present examples of valid use cases. A user can phrase a command for the agent in various ways. Moreover, the agent can converse with the user if it recognizes an intent that information is missing. For example if the user commands the agent to "turn the lights", the agent will reply with either "where?" or "in which room?". If the user replies with a valid room (i.e., "In the kitchen") the agent will reply "Do you want them on or off?". If the user replies with either "on" or "off" the agent will execute the *turn-lights action*. A representative example of the generative process of the

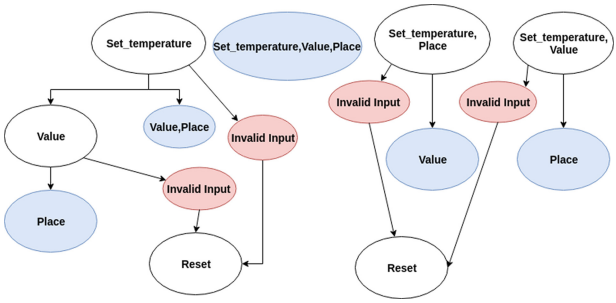

Fig. 1. Finite automata of story generator example

algorithm through the conversational flow between the agent and a user, as far as set temperature-single intent command are concerned:

- Active node = stands on an active state that has not fulfilled all requirements and is able to generate new scenarios
- Final node = stands on a final state and cannot generate new scenarios

A case example was used as part of the evaluation process. The example aims to demonstrate that the Novel Dialogue Generator of CIPA–gen A is able to produce all the possible use case scenarios in its conversation with a user by adopting the concept of adjacency pairs. In particular, in the presented example the agent requires information about the value of the temperature and the place (room). There are 4 initial cases: (a) the user defines the intent accompanied by the value or (b) the place or (c) both of them or (d) the case that the user inputs only the intent. In the next step of this discourse, the user can either provide valid information that fulfills a slot or provide an invalid input that is categorized in one of the 14 invalid operations in the sector of energy (lights, hvac, etc) including no-understand operation in the case of no relevant input. Every single invalid input in the Fig. 1 diagram is translated to multiple alternative rejected scenarios. The node that contains value and place is a final node. On the contrary, the rest of the nodes need at least one more slot to be fulfilled in order to end up in a final state. In the same logic, every active node generates more complex scenarios until a final node is born. In this example, the *set_temperature* intent generates 64 unique scenarios that cover all possible contingencies and they are illustrated from the colourful nodes in the Fig. 1. Each node represents the given input from the user. Calculation of all possible generated scenarios in a multi intent command is a process identical to the single intent, differentiating only in the number of slots. Multi intents commands have as slots the union of the slots of two intents. So, for example in the *set_temperature_lights* intent that is responsible to set the temperature in a room and turn the lights, the available slots comprise the union of (value, place) from *set_temperature* and $lights\_on\_off$ from light. The total number of generated scenarios are 281.

The evaluation of CIPA story generator was an experimental process that conducted through a comparison between the enumeration of all possible scenarios that can be generated from a dialogue among an agent and a user as illustrated in the Fig. 1 and the actual scenarios that are generated from CIPA (Parsed data from the generated file and calculated the sum of generated scenarios for every intent). CIPA is evaluated as a competent agent that behaves properly covering all contingencies.

5 Conclusions and Future Work

In conclusion, this paper introduces a general purpose task-based agent, equipped with a novel dialogue generator, and it was applied to two domains of the CERTH *nZEB Smart Home*, energy and health domain. The developed CIPA–gen A is based on the RASA framework and supports multi-intent for

the aforementioned domains capable of recognizing up to two intents per user command. Towards to a domain agnostic agent, the code-base was engineered so as to be easily applied to different domains. CIPA–gen A uses an embedding method for its NLU model that is not based on pre-trained word vectors of a specific language so as it can be generalized to *any* natural language. The training data should have to be generated for that language, using the tools selected and the algorithms developed, while omitting a dependency on SpaCy for slot extraction. This consists the only limitation on training a model for a different language. In addition, messages from real-world users should be collected in order to build a real usage data set that can be used for both training the model to achieve greater accuracy and for evaluation purposes. Furthermore, a novel dialogue generator, based on the idea of adjacency pairs, has been implemented in order to enable the proposed agent to generate all the possible scenarios in a conversation between the agent and a user.

Future research and development related to CIPA agent will be focused on the extension of the agent's capabilities. A first enhancement would be the replacement of the LSTM of the dialogue model with a differential neural computer (DNC) [4]. Next releases of the agent will support complex objectives in its action model, i.e., the agent will be able to form a complex plan of action based on its conversation with the user and execute it and by utilizing AI Planning research [6]. The new system will be the CERTH Advanced Multi-Domain Reasoning and Planning System (CAMDRaPS). This idea has been considered before by MIT researchers [5] but it has not been developed yet. In general, it is planned to add an open-source classical planner, more advanced conversion model from intents/entities to planning actions/variables, machine learning techniques to improve classification of the domain, Probabilistic Planning and Probabilistic Reasoning over Time.

Acknowledgements. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 643607 (myAirCoach) & No 732679 (ACTIVAGE).

References

1. Weizenbaum, J.: ELIZA a computer program for the study of natural language communication between man and machine. Commun. ACM **9**, 36–45 (1996). https://doi.org/10.1145/365153.365168
2. Dersch, W.C.: Shoebox - a voice responsive machine. Datamation **8**(6), 47–50 (1962)
3. Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., Weston, J.: StarSpace: embed all the things!. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
4. Graves, A., et al.: Hybrid computing using a neural network with dynamic external memory. Nature **538**(7626), 471–476 (2016)

5. Yu, P., Shen, J., Yeh, P.Z., Williams, B.: Towards personal assistants that can help users plan. In: Traum, D., Swartout, W., Khooshabeh, P., Kopp, S., Scherer, S., Leuski, A. (eds.) IVA 2016. LNCS (LNAI), vol. 10011, pp. 424–428. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47665-0_47
6. Nau, D., Ghallab, M., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann Publishers Inc., San Francisco (2004)
7. Midgley, T.D., Harrison, S., MacNish, C.: Empirical verification of adjacency pairs using dialogue segmentation. In: Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, pp. 104–108. Association for Computational Linguistics (2006)
8. Goddeau, D., Meng, H., Polifroni, J., Seneff, S., Busayapongchai, S.: A form-based dialogue manager for spoken language applications. In: Proceedings of the International Conference on Spoken Language Processing, Philadelphia, PA, USA, 3–6 October 1996, pp. 701–704 (1996)
9. Traum, D.R., Larsson, S.: The information state approach to dialogue management. In: van Kuppevelt, J., Smith, R.W. (eds.) Current and New Directions in Discourse and Dialogue. Text, Speech and Language Technology, vol. 22. Springer, Dordrecht (2003). https://doi.org/10.1007/978-94-010-0019-2_15
10. Dumitrescu, S.D.: Cassandra smart-home system description. In: 2017 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Bucharest, pp. 1–6 (2017)
11. Serban, I.V., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A.C, Bengio, Y.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: AAAI, pp. 3295–3301 (2017)
12. Park, Y., Cho, J., Kim, G.: A hierarchical latent structure for variational conversation modeling. ArXiv, abs/1804.03424 (2018)
13. Vinyals, O., Le, Q.: A neural conversational model. In: Proceedings of ICML Deep Learning Workshop (2015)
14. Li, J., Galley, M., Brockett, C., Spithourakis, G.P., Gao, J., Dolan, B.: A persona-based neural conversation model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, pp. 994–1003 (2016)
15. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., Jurafsky, D.: Deep reinforcement learning for dialogue generation (2016)
16. Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S.: A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. Knowl. Eng. Rev. **21**(02), 97–126 (2006)
17. Park, Y., Kang, S., Seo, J.: An efficient framework for development of task-oriented dialog systems in a smart home environment. Sensors (Basel) **18**(5), 1581 (2018). https://doi.org/10.3390/s18051581
18. Schegloff, E.A., Sacks, H.: Opening up closings. Semiotica **8**(4), 289–327 (1973)