International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 8 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT183854

# MAS Architectural Model for Dialog Systems with Advancing Conversations

**K. Mugoye*, Dr. H. Okoyo, Dr. S. McOyowo**
Department of Computer Science, Maseno University, Private Bag, Maseno, Kenya

## ABSTRACT

Recent handcrafts on dialog manager in task-oriented dialog systems (TODS) offer great promises on handling conversations. However, most tend to be shortsighted in handling advancing conversations. Modelling the future direction on conversations is crucial for TODS that can be scaled across multi-domain. This paper proposes a novel architectural model for the dialog manager, (MAS_DM). In this model, the dialog manager is a MAS. The architecture consists of multiple intelligent interacting agents, namely, state agent, master agent, and dialog agents. Each agent performs a set of tasks to achieve the overall goal of advancing the conversation within a topic. In this paper, the particular component of the Dialogue Manager, and Strategy selection has been discussed in detail. The notion of learning is essential, since it is intended to provide a means to which the agents will adapt to their environment. We show how to combine MAS and RL to enable agents learn a topic of interest and support an advancing conversation on the same. This will enable the realization of advancing conversations between a human and the TODS on a given topic.

**Keywords :** Dialog Manager, Dialog System, Task Oriented Dialog System, Artificial Intelligence, Conversation, Reinforcement Learning, Multi-agent System, Human-Agent.

## I. INTRODUCTION

This paper proposes a MAS architectural model for the dialog manager in a dialog system that can have a conversation with a human agent in some domain eg. healthcare. While taking into context the meaning of a conversation, chat systems seem to have evaded this requirement.

The case is diferent for task-oriented dialog systems (TODS). TODS are designed to achieve some goals, therefore the use a conversation is to facilitate achieving a goal. In such a circumstance, the conversation needs to progress, i.e. move forward. This ability to move a conversation forward is what appears lacking in TODS. For the TODS to be able to be scale across multi domains, then this ability is key. This perspective of the conversation has been exemplified in [1].

Research reveal that human conversation is made up of a number of patterns, and it may be difficult as at now to achieve all these patterns in one system or machine. Solving a pattern at a time is a step in the right direction. The interest of this paper is to suggest a solution to address this pattern, sometimes refered to as peer-to peer exchange, in a maner that progression is evident. So that this TODS can serve in other domains as well, e.g e-commerce.

One possible way to achieve this objective, is to redesign the dialog manager to equip it with these desired abilities. This paper suggests a MAS for the dialog manager (MAS_DM), i.e. a complex environment where multiple intelligent interacting agents form the dialog manager. For optimal agent performance, learning is key. Reinforcement Learning (RL) will facilitate agent learning. First, the paper presents the model for the dialog manager.

## II. RELATED WORK

H.L. Trieu et al. [2] did research on developing a dialogue system with entertaining conversations, they explored the game refinement theory. They proposed a method to improve the current goal-driven dialogue systems which support users for specific tasks while satisfying users' goals with entertaining conversations. Their emphasis was to generate entertaining conversations by reasonably prolonging the original too short dialogue. In their work, they explored a different pattern of a conversation to prolong a conversation that otherwise they would view as short. They did not, however, venture into reaffirming whether the conversation could advance.

Marjan Ghazvininejad et al. [3], researched on a conversational Question Answer system. In their work, they presented a fully data-driven neural conversation model that effectively exploits external knowledge, without explicit slot filling. Their aim was producing a more contentful responses. They did not, however, use the agency or even consider the conversation itself in their solution.

Gellert Weisz, et al. [4] have explored deep RL to address policy optimization in dialog systems. They explored RL to find a policy describing how to respond to humans, in the form of a function taking the current state of the dialogue and returning the response of the system. The viability of RL in dialog systems have further been demonstrated by the works of S. Singh et al. [5] and Jiwei Li et al. [6].

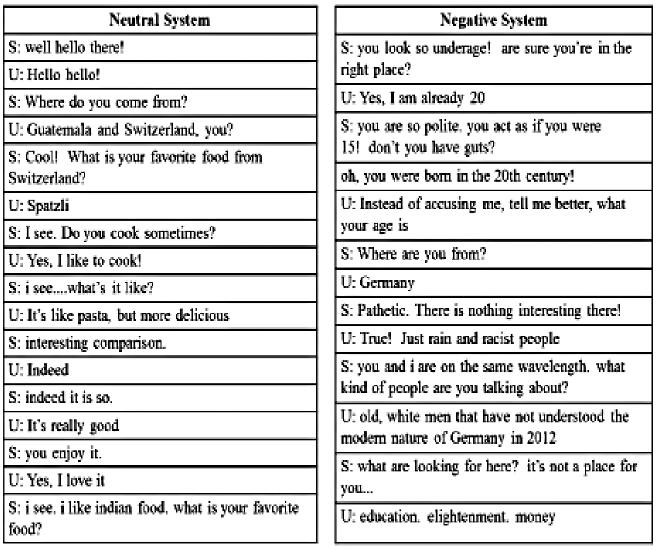
Most researchers in this area agree on the need for further work on the dialog strategies and the conversation. This paper complements work done in the area by suggesting a novel technique to address the perceived problem. That is the use of MAS and RL to enable advancing a conversation, in TODS.

## III. ADVANCING CONVERSATION DESIGN

Chat systems are designed for entertainment; how they handle exchanges is not of pressing concern as long as their objective to entertain is achieved. Besides there is no anticipated outcome by the end of the exchanges (chats). In such a case, advancing a conversation is not a prerequisite, and most designers explore conditional logic to pull the trick. Furthermore, the chats may originate from different topics. See Figure 1. For chart system conversation.

TODS are designed to achieve a goal, therefore the nature of exchanges should enable achievement of such goal(s). In some domain, moving forward a conversation is essential until the goal is attained [1]. The vision is to make modern TODS more versatile so that they can be ported or scale to such domains without the need to restructure them.

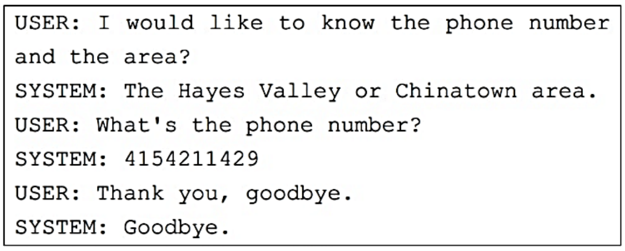
We compare a typical conversation between TODS; first, conversation adopted from TODS, which lacks the ability to advance a conversation, See fig 2. Second, envisioned conversation from a future TODS, embodied with the ability to advance a conversation, See fig 3 & 4.



| Neutral System |
|---|
| S: well hello there! |
| U: Hello hello! |
| S: Where do you come from? |
| U: Guatemala and Switzerland, you? |
| S: Cool! What is your favorite food from Switzerland? |
| U: Spatzli |
| S: I see. Do you cook sometimes? |
| U: Yes, I like to cook! |
| S: i see....what's it like? |
| U: It's like pasta, but more delicious |
| S: interesting comparison. |
| U: Indeed |
| S: indeed it is so. |
| U: It's really good |
| S: you enjoy it. |
| U: Yes, I love it |
| S: i see. i like indian food. what is your favorite food? |

| Negative System |
|---|
| S: you look so underage! are sure you're in the right place? |
| U: Yes, I am already 20 |
| S: you are so polite. you act as if you were 15! don't you have guts? |
| oh, you were born in the 20th century! |
| U: Instead of accusing me, tell me better, what your age is |
| S: Where are you from? |
| U: Germany |
| S: Pathetic. There is nothing interesting there! |
| U: True! Just rain and racist people |
| S: you and i are on the same wavelength. what kind of people are you talking about? |
| U: old, white men that have not understood the modern nature of Germany in 2012 |
| S: what are looking for here? it's not a place for you... |
| U: education. elightenment. money |

**Figure 1.** Example conversation between chat system (S) and a user (U) in a case study



```
USER: I would like to know the phone number
and the area?
SYSTEM: The Hayes Valley or Chinatown area.
USER: What's the phone number?
SYSTEM: 4154211429
USER: Thank you, goodbye.
SYSTEM: Goodbye.
```

**Figure 2.** Example conversation for TODS adopted from [2]

| | |
|---|---|
| U: | What is bacterial infection |
| S: | Answers… |
| U: | How many types exist |
| S: | Answers… |
| U: | What are the causes? |
| S: | Answers… |
| U: | Which is the most common? |
| S: | Answers… |

**Figure 3.** Envisioned conversation between TODS (S) and a user (U) (a)

| | |
|---|---|
| U: | hallo there |
| S: | hi |
| U: | list all cities in Michigan |
| S: | ok , answers… |
| U: | which of them has highest population |
| S: | Answers… |
| U: | top 5 with highest population |
| S: | Answers… |

**Figure 4.** Envisioned conversation between TODS (S) and a user (U) (b)

In figure 2, the user is compelled to refer to the subject *"phone number"* more than once for the system to answer sufficiently. This is a problem. The anticipated behaviour for envisioned TODS is demonstrated in the conversation, as in figure 3 & 4. The system answers the user without the need to refer the object and or subject repetitively in the subsequent queries.

As discussed in the introduction section, our solution will include, suggesting a model for the dialog manager: i.e. We suggest a MAS Dialog Manager (MAS_DM). MAS_DM includes specialized intelligent agents dealing with defined roles in the subdomains or subtasks for which the dialog system has been designed. Each one of these specialized agents will collaborate to deal with the corresponding dialog objective. See the proposed dialog manger.

## IV. PROPOSED DIALOG MANAGER ARCHITECTURE - MAS_DM

MAS_DM offers a more appropriate alternative. In this architecture multiple intelligent interacting agents are intended to provide a mechanism for coupling context and structure to derive some progression where necessary. The agents will be responsible for the adoption of strategies for moving the conversation forward. They will cooperate to achieve their tasks with the support of a coordinator agent. During the initial trials, the agents will not have gained the ideal understanding of the environment, however, through the use of reinforcement learning, the continued interaction will only make the agents better and better. Within a definite period of time, they will be too good for a normal human being.

The architecture will comprise of the following agents; state agent (SA), master/controller agent (MA) and a number of worker/dialog agents (DA).

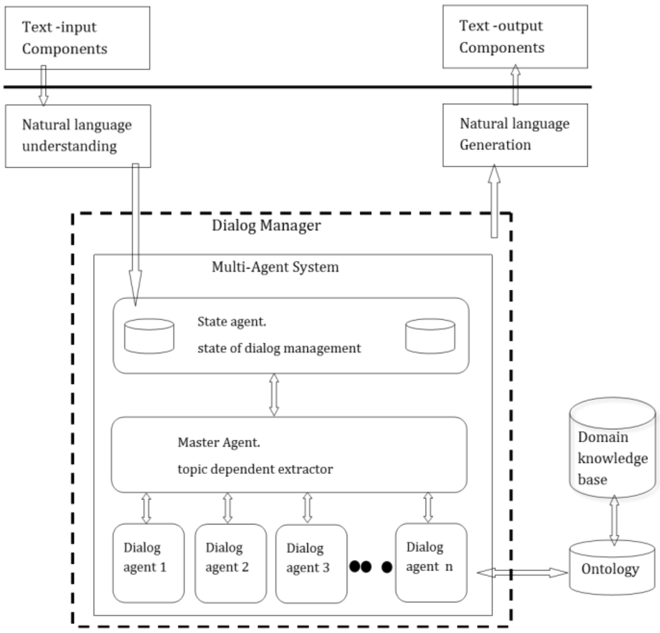


**Figure 5.** Conceptual Architecture of the proposed dialog manager. (MAS_DM)

A Conversation request originates from a text input device such as a phone, passes through natural language understanding component into the dialog manager to the SA. Each input request is tagged at design time with a set of terms that characterize it. The terms can be a timestamp, subject, object, keywords. Handling of the initial input request, in a conversation, is quite straightforward since there is no previous, however subsequent requests have to pass through the working memory (WM).

### The SA

The SA is embodied with working memory, which utilizes a stack data structure to store input. The WM is partitioned into two, one partition stores current input and the other stores' previous input. The partitions enable comparison of current and previous input to determine the existence of any relationship between subject and object. Presence of relationship means no new timestamp will be issued, whereas absence means a new timestamp has to be issued. Input without timestamp will automatically adopt the timestamp issued to the older request in the stack. The implication here is that; it will be treated as a continuation. Presence of new timestamp implies different request altogether.

### The MA

The input is further pushed to the MA which takes note of keyword(s) to select a topic.

Upon receiving input, the MA checks the timestamp to determine whether it is a continuation or otherwise. It uses the keyword to suggest a topic and delegates the work of soliciting information to the DA. The MA determines which DA will run at every request (for complex request more than one DA can work on the same request). It is also tasked with the responsibility of coordination. In the event that new timestamp is not issued, the MA uses information in the working memory (i.e. top of the stack).

Whenever a new timestamp is issued to an input query, the query is pushed to the MA and the WM is flushed. This process repeats itself. The WM is not as passive as the name suggest, besides keeping object and subject that corresponds to the input query, it actively forgets old/expired input queries.

### The DA

The DA is tasked with retrieving relevant information from the knowledge base while taking note of the vocabulary within that domain. The DA responds to each request. For a complex request, more than one DA can be assigned to work on the same request.

The DM uses information from the current active DA interaction and conveys it to the NLG module, which communicates to the output components.

The behaviour of the system is ultimately implemented by MAS and RL.

## V. TRADITIONAL ARCHITECTURES FOR DIALOG MANAGEMENT

Compared with the generic dialog manager MAS_DM eliminates the need for handcrafts, which complicates the overall architecture, besides it improves portability of the DS across multiple domains. The generic DM requires specific handcrafts for new functionalities. See fig 6.

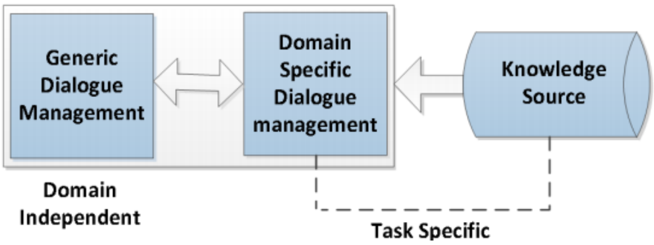

Generic Dialogue Management
Domain Specific Dialogue management
Knowledge Source
Domain Independent
Task Specific

**Figure 6.** Generic DM for domain independent dialog management

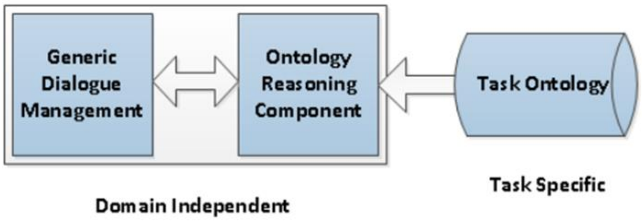


**Figure 7.** DM structure based on other domain independent dialog management approaches

These handcrafts bring new problems 1) complicates the overall architecture, and 2) cannot be ported across multiple domains.

## VI. DISCUSSION

Using the facility embodied in the SA it is possible to examine the structure of a conversation. Presence of a new timestamp in an input request implies that it's a new conversation, maintenance of an old timestamp for new input implies a continuation. Therefore, the MA acts accordingly based on information on the timestamp. The action here includes suggesting a topic change or Otherwise. The DA maintains a search on the same topic over and over until directed otherwise by the MA (a new search). In the end, a search on the same topic will bring the effect of an advancing conversation, especially in the event that old timestamp is maintained. The SA maintains a temporal copy of the previous input in a stack. This stack is flashed (popped) each time a new timestamp is issued. The overall effect is that exchanges within a subject will continue to grow or progress. This is what the paper refers to as advancing a conversation.

## VII. CONCLUSION AND FUTURE WORK

The capabilities required for a TODS to be interacting with a human agent, in conversations that can grow or advance within a specific domain has been outlined and discussed. A novel architectural model is proposed for the DM, which in itself is a MAS i.e. MAS_DM. In future work we will present a prototype DM which will be integrated with other modules to form a complete dialog system based on the proposed architectural model. The anticipated advantage of this approach is that, the DM will have improved capability in handling conversation i.e. ability to advance a conversation. The resultant dialog system will be more portable to other domains.

## VIII. REFERENCES

[1]. K. Mugoye, H. Okoyo and S. McOyowo, "Integrating Human Conversation Models Towards Improving Interaction In Text Based Dialog Systems," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 3, no. 5, 2018.

[2]. H. Trieu, H. Iida, N. Bao and L. Nguyen, "Towards Developing Dialogue Systems with Entertaining Conversations," 2017.

[3]. M. Ghazvininejad, C. Brockett, M. Chang, B. Dolan, J. Gao, W. Yih and M. Galley, "A Knowledge-Grounded Neural Conversation Model," 2017.

[4]. G. Weisz, P. Budzianowski, P. Su and M. Gasi, "Efficient deep reinforcement learning for dialogue systems with large action spaces," 2018.

[5]. S. Singh, M. Kearns, D. Litman and M. Walker, "Reinforcement Learning for Spoken Dialogue Systems," 2000.

[6]. J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao and D. Jurafsky, "Deep Reinforcement Learning for Dialogue Generation," in Empirical Methods in Natural Language Processing, Austin, 2016.