International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 5 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT19558

Towards Logically Progressive Dialog for Future TODS to Serve in Complex Domains

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

ABSTRACT

Complex domains demand task-oriented dialog system (TODS) to be able to reason and engage with humans in dialog and in information retrieval. This may require contemporary dialog systems to have improved conversation handling capabilities. One stating point is supporting conversations which logically advances, such that they could be able to handle sub dialogs meant to elicit more information, within a topic. This paper presents some findings on the research that has been carried out by the authors with regard to highlighting this problem and suggesting a possible solution. A solution which intended to minimize heavy reliance on handcrafts which have varying challenges. The study discusses an experiment for evaluating a novel architecture envisioned to improve this conversational requirement. The experiment results clearly depict the extent to which we have achieved this desired progression, the underlying effects to users and the potential implications to application. The study recommends combining Agency and Reinforcement learning to deliver the solution and could guide future studies towards achieving even more natural conversations.

Keywords : AI Chatbot, dialog system (DS), logical progression in conversation, chat-oriented dialog system, task-oriented dialog system, Reinforcement learning

I. INTRODUCTION

Task-Oriented dialog systems (TODSs) are designed specifically to help users achieve a task within a closed domain. The last half a decade has seen their applications continue to grow and also the emergence of new domains seeking to profit from their use. This however brought new challenges. To flourish in some of these new domains, new demands have to be met. Take for instance domains like complex information retrieval and question answering. More is demanded.
Directional flow type of dialog can no longer hold, but instead, efforts towards natural conversation seems to offer more promise. Research however show that there are many pattern in a full natural conversation and that we are far from achieving that, but addressing any pattern is a right step. This paper featured an experiment of testing a dialog system (DS) commonly referred to as AI-Chatbot prototype which could offer solution to logically advancing conversation.

Three such recent studies were conducted by Mugoye, Okoyo and McOyowo [1, 2, 3], in a move to understand human to human conversation, human to machine conversation so as to highlight the missing pattern in human-machine conversation.

The first study [1] characterized human conversation so as to pin point the missing pattern in human machine conversation. It featured different models in communication and how we can map a model towards designing interfaces to achieve better interaction results. It focused on considering usability issues during the designing of interactive systems for making better and usable systems. This study was limited to making task oriented dialog systems, reduce memory load from users and provide easy, enjoyable interaction, by allowing progressive search during information retrieval.

The second study [2] presented a method and an architectural model that could lead to offering a solution with respect to the desired logical progression in a conversation, while considering extensibility in the future.

The model advocated for agency approach, where intelligent agents are equipped with mechanisms to understands structure in or within sentences, take note of the conversation context and user intentions. Further, machine learning module, which could be regarded as an agent too, depending on the implementation platform, participates in action selection, sometimes referred to as policy selection by other sources. In the end, the result is a product of joint participation of all these intelligent entities. We anticipated to profit from the capabilities and advantages of agency. The third study [3] demonstrated a real application of the solution to address some maternal healthcare challenges in Kenya. Demonstrated practicability in maternal healthcare domain.

The theory and efforts in the studies [1, 2 and 3] complemented each other, however the study would be more complete if its practicality is tested through a running prototype.

The rest of the paper is organized as follows: section II presents some effective method and materials / outlines the research design and methodology used, section III presents the experiments and evaluation, while section IV discusses experiment results. Conclusion and future work are given in the final section.

II. METHODS AND MATERIAL

The construction of our prototype required; a Platform Tool, the dialog management architecture (DMA) [2], and adapting the DMA to the Platform Tool [4]. Adaptation of the DMA to a specific Platform Tool required detailed knowledge on how the tool is implemented, even though this is essential, it is however not a goal in our study. Figure 1 presents a high level diagram of the architecture of our prototype.

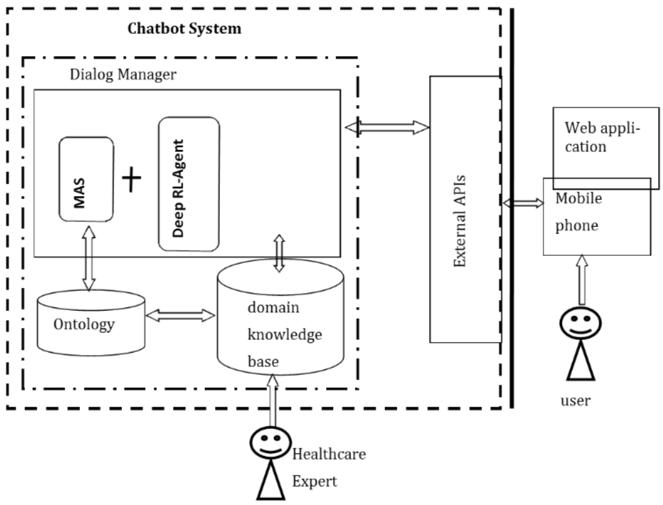


Figure 1. High level diagram of the prototype architecture

The idea and choice of our platform tool was informed by considering a number of essential factors. One, supports for agency; two, adequate libraries for reinforcement learning; three, ability to integrate a knowledge base and other resources; four, support for deployment. The tool which met most of our essential requirement was dialogflow [4]. Despite, having available documentation, the version within our reach had its limitations. We overcame some of these limitations by customizing some functionality in the toolkit.

Adaptation to the platform involved customizing the functionalities which were not directly provided by the tool, and crafting of the desired behaviour by the entities. A discussion on the same is provided in this section. We first created two homogeneous agents in different projects and equipped each with some basic but distinct functionality. Basic here referred to sufficient for the purpose of the study. Second, we stretched the import feature to load another agent in the project, changing the composition of agents to: the original and loaded agents. Since this functionality is not supported by the toolkit, we faced two challenges. One, the intents of the main agent intents were overridden by the loaded agent, and two, there was conflict or confusion in handling of contexts. In our approach, we had to make distinct the intents of the loaded agent, so that, the intents of the original agent are not overridden. We suppressed conflicting context from the loaded agent and mapped other context to the preferred context of the original agent, to enable both agents relate to similar context.

This Customization is both programmatically and through platform interface. For each agent, a session_id was generated to uniquely identify the agent. We distinguished the intents of each agent by attaching the agent's session_id to the intent. We implemented the logic which systematically calls and maps the agents to their intents. Figure 2, show a code snippet of how intents from different agents could be managed, at runtime.

```
import dialogflow
def detect_intent_texts(project_id,session_id,text)
    session_client=dialogflow.SessionClient()
    session=session_client.session_path(project_id,session_id)
    text_input=dialogflow.types.TextInput(text=text)
    query_input=dialogflow.types.QueryInput(text=text_input)

    response=session_client.detect_intent(
    session=session,query_input=query_input)

session_id= initialize_session_id_from_application
text=get_text_from_application

if condition==1:
    res=detect_intent_texts(project_id1,session_id,text)
elif condition==2:
    res=detect_intent_texts(project_id2,session_id,text)
else:
    res=detect_intent_texts(project_id3,session_id,text)
```

Figure 2. Intent management for multiple agents, code snippet in python.

The implication here was that a query within a particular context could be answered by either agent depending on the depth of requested information. The agents could solicit more information from the user query as they build up a response. In summary, the responses, suggestions or advice were a collective contribution of the agents.

III.EXPERIMENT AND EVALUATION

The prototype, AI Chatbot, named Mshauri Wako, was available online for beta test for a period of 31 days, each tester was required to try it at least three times before filling a survey. The survey was configured to be taken only once for every user. Data obtained was coded based on calibrations on table 1. This data was used to generate the confusion matrix M, and was adopted in our hybrid model of evaluation.

We identified attributes relevant to the study, and featured PARADISE [5] and GQM [6] evaluation models. The model [5], include the use of the Kappa coefficient [7] and [8] to operationalize task success, and the use of linear regression to quantify the relative contribution of the success and cost factors to user satisfaction.

The identified attributes were classified in reference to ISO 9241. We created PARADISE-based objectives which were mapped directly to the task success and dialog performance objectives suitable for our Chatbot evaluation. Table 1 depicts the selected metrics, within PARADISE.

Table 1. Selected metrics for our Chatbot aligned in terms of ISO 9241.

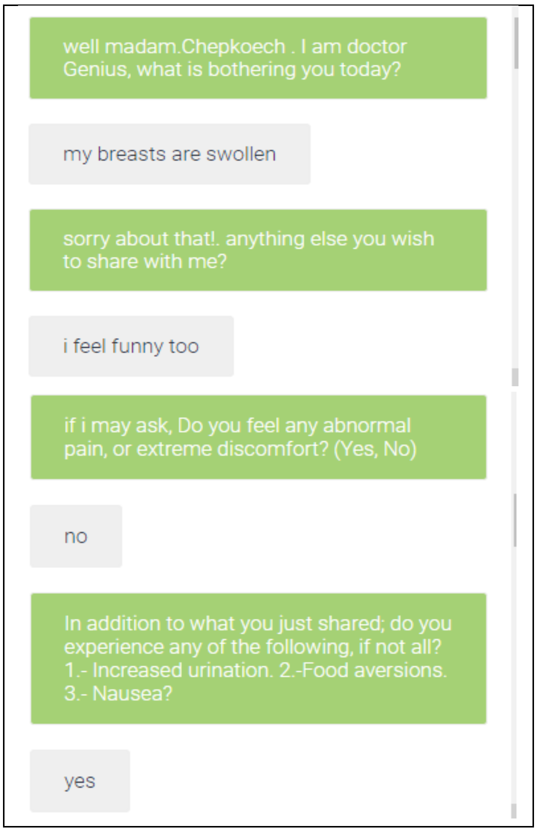
Quality Attribute	Category	Reference
Satisfaction		
• Can detect meaning / intent	Accessibility	Wilson et al. [9]
• Convey personality • Provide greetings • Make task more fun	Affect	Morrissey & Kirakowski [10] Eeuwen [11]
Effectiveness		
• Accuracy of Concept	Functionality	
• Maintain satisfying, natural interaction		Morrissey & Kirakowski [10]
• Interpret utterances correctly • Able to maintain themed discussion	Humanity	Eeuwen [11]
Presentation of knowledge and additional functionality		
• Able to refer to external sources	Knowledge	Cohen & Lane [12]

A. Tasks as Attribute Value Matrices

We used attribute value matrix (AVM), table 2, to represent dialogue tasks. AVM consists of the information that must be exchanged between the agent and the user during the dialogue, represented as a set of ordered pairs of attributes and their possible values. Figure 3 shows a sample conversation from Mshauri Wako Bot.

Table 2. Our AVM instantiation, scenario keys

Attribute	Actual value (sample)
Accessibility (AC) Affect (AF) Functionality (FX) Humanity (H) No of user Utterances (NUU)	Detect an intent, sentence A greeting or a bye Give relevant information Correct interpretation of context No of utterances



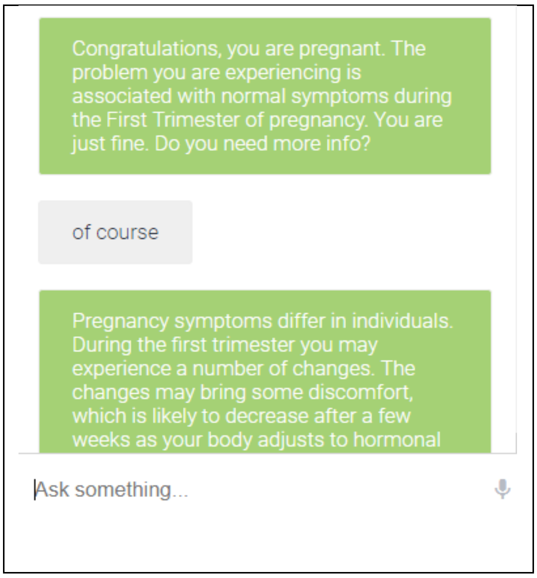


Figure 3. Fragment of a progressive conversation- from Mshauri Wako BOT.

B. Measuring Tasks Success

We measured task success for a whole dialogue by how well the agent and user achieve the requirements of the task by the end of the dialogue. The matrix M, in figure 4 shows in summary how the 60 AVMs representing each dialogue with our Chatbot compare with the AVMs representing the relevant scenario keys, where applicable.

N = 60		Greetings				Names				User Problem				System Response				More Information			
	DATA	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
	V1	12			1																
	V2		26		1																
	V3			10	1																
Greetings	V4			1	8																
	V5					4															
	V6						10														
	V7							13													
Names	V8						1		32												
	V9									3											
	V10										13										
User	V11									1		14									
Problem	V12												29								
	V13													3	2						
	V14														2						
System	V15													1		33					
Response	V16													2	2		14				
	V17																1	3		1	
	V18																		10		1
More	V19																	1		30	
Information	V20																			1	13
SUM		12	26	11	11	4	11	13	32	4	13	14	29	6	6	33	15	4	10	32	14

Figure 4. Confusion matrix, M.

Labels v1 to v4 represent the possible values related to greetings, v5 to v8 represent possible values of related to names, v9 to v12 represent possible values related to User Problem, v13 to v16 represent possible values related to System Response, v17 to v20 represent possible values related to More Information, in each matrix. Columns represent the key, specifying the information values the agent and user were supposed to communicate to one another given a particular scenario. The blanks in columns suggest we did not have to offer guidance on further response.

Given our AVM and matrix M, we compute $P(E)$ and $P(A)$ by applying Equation (4.2) and (4.3) respectively. We obtain a $P(E)$ of 0.061; and a $P(A)$ of 0.940. We apply Equation (4.1) to obtain a (K) of 0.936.

Kappa coefficient, defined in equation 1.

$$\mathrm{K} = \frac{P(A)-P(E)}{1-P(E)} \qquad \text{…………… Equation (1)}$$

Where, P(A) is the proportion of correct interpretations, and P(E) is the correct interpretations occurring by chance. Since in our case, the prior distribution of the categories is unknown, we estimate P(E), from the distribution of the values in the keys. As in equation 2.

$$P(E) = \sum_{i=1}^{n} \left(\frac{ti}{T}\right)^2 \qquad \text{………..Equation (2)}$$

where (ti) is the sum of the frequencies in column (i) of M, and T is the sum of the frequencies in $M = (ti + \cdots + tn)$.

P(A), is always computed using formula in equation 3.

$$P(A) = \sum_{i=1}^{n} \left(\frac{M(i,i)}{T}\right) \qquad \text{………Equation (3)}$$

Next we measured the systems performance.

C. Estimating a Performance Function

The overall performance is computed as in equation 4.

$$p = (\alpha * N(\mathrm{K})) - \sum_{i=1}^{n} wi * \mathrm{N}(ci) \qquad \text{………..Equation (4)}$$

Where N is a Z score normalization function, α is a weight on (K), and the cost function (ci) are weighted by wi. Here, we used N to overcome the problem where values of (ci) , which may be calculated over widely varying scales ,are not on the same scale as (K). This is a problem normally addressed by normalizing each factor x to its Z score as in equation 5:

$$\mathrm{N}(x) = \frac{(x-\bar{x})}{\sigma_x} \quad \text{…………………Equation (5)}$$

Where σ_x is the standard deviation for x.

To determine the systems performance, we tagged all the AVM attributes with respective costs. Our cost attributes comprised of: *AF, FX, H* and *NUU*. The attribute *NUU* which qualified as our (ci) was in a different scale, therefore, we applied Equation (5) for normalization. In the next step, we apply Equation (4), however, the equation will not be complete if the values for the weights α and wi are unknown. Here, linear regression is used for this purpose.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.940151756
R Square	0.883885324
Adjusted R Square	0.877664895
Standard Error	0.31128839
Observations	60

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	41.30690748	13.76896916	142.093948	3.73938E-26
Residual	56	5.426425853	0.096900462		
Total	59	46.73333333			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.689791309	0.266606462	2.587301541	0.012296351	0.155714389	1.223868229	0.155714389	1.223868229
AF	0.28147159	0.086972683	3.236321812	0.002035315	0.107244371	0.45569881	0.107244371	0.45569881
FX	-0.058345803	0.111901417	-0.521403607	0.604141669	-0.282511278	0.165819673	-0.282511278	0.165819673
H	0.651945325	0.114614295	5.688167676	4.86039E-07	0.422345304	0.881545347	0.422345304	0.881545347

Figure 5. Regression Output-1

Figure 5 shows the overall contribution of our attributes as statistically significant. However, individual contribution show FX is not statistically significant. For this reason, we eliminate attribute (FX) and perform a second linear regression. We obtain the results as in Figure 6.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.939851917
R Square	0.883321625
Adjusted R Square	0.879227647
Standard Error	0.309293745
Observations	60

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	41.28056395	20.64028198	215.7612015	2.56582E-27
Residual	57	5.452769379	0.095662621		
Total	59	46.73333333			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.718443699	0.25921	2.771666596	0.007519615	0.199384627	1.237502771	0.199384627	1.237502771
AF	0.250955245	0.063921296	3.926003686	0.000235317	0.122955057	0.378955433	0.122955057	0.378955433
H	0.622387613	0.098975077	6.288326613	4.82791E-08	0.42419344	0.820581787	0.42419344	0.820581787

Figure 6. Regression Output-2

This regression produces coefficients or weights describing the relative contribution of predictor factors accounting for the variance in a predicted factor. We sum the coefficients to obtain wi of 0.8733; we note the intercept 0.7184 which forms our α .

To obtain overall performance we get the average ci. We obtain the average NUU as 22.567, which becomes 23 to the nearest integer.

We obtain the mean, $\bar{x}$ = 14.1 and σ_x = 10.279771, therefore, $N(\mathrm{x})$ where x is *NUU*, is applied on Equation (4.5) to get $N(\mathrm{K})$=0.87(0.936).

Now we have both $N(\mathrm{K})$=0.814 and $N(ci)$=0.046,

We apply Equation (4)

= (0.7184* $N(K)$) - 0.8733* N (ci)

p = (0.7184*(0.87*0.936)) - (0.8733*(0.046)) = 0.545

p = 54.5 % (as a percentage)

D. GQM Evaluation

First, we refined the stated goals into a set of quantifiable questions. This set of questions were then used to identify relevant data to be collected, and guided the selection of appropriate metrics. The data collected here is used for decision making, and to analyze whether the defined goals had been achieved. Tables 3 and 4 describe the goals and metrics based on the model, response column show the results after analysis.

Table 3. GQM description customized for our purpose

			response
Goal 1	Purpose Issue Object Viewpoint	Implement a DS that support Logically progressing Conversation From the user's viewpoint	**79.72**
Question	Q1	Is the DS advancing a conversation?	
Metrics	M1 M2 M3	-Support of Sub-dialog to feed into main dialog -Occurrence of progressive exchange - % Number of correct responses	Yes Yes 93
Question	Q2	Are user satisfied?	
Metrics	M4 M5	-% Ease of interaction -% Enjoyability of interaction	75 80
Question	Q3	Is the architecture suitable for advancing conversation?	
Metrics	M6 M7	-% Realization of conversation goal -% Naturalness of conversation	78 72

Table 4. GQM description customized for our purpose

			response
Goal 2	Purpose Issue Object Viewpoint	Verify if the DS informatively handles the conversation from the user's viewpoint	**86.8**
Question	Q1	Is the exchange relevant to a user query?	
Metrics	M1 M2 M3	Classification of the exchanges User perception of the conversation Number of correct responses	2 Enjoyable 0.93
Question	Q2	Does the exchange elicit more information about the query?	
Metrics	M4	User willingness to use system again	80

IV. RESULTS AND DISCUSSION

Our results from the first evaluation, demonstrated that our AI Chatbot conversations achieved 0.94 correct interpretations and an estimate of 0.061 correct interpretations occurring by chance. Thus yielding a task success rate of 0.936 and an overall performance score of 0.545. Further results from second evaluation demonstrated two things. First, a usability score of 83.26%, second, (1) the prototype supported logically progressive exchanges to handle sub dialogs meant to elicit more information, and (2) provided an enjoyable interaction.

We present a novel architecture, and method along with the implementation of a running prototype. Generally, our architecture obtained good results: - besides making the conversation more natural, the architecture brings several benefits. First of all, it decouples dialog context tracking and complex dialog control into individual segments: - this simplifies maintenance. Second, it did not set any boundaries on how more functionality can be added: -this is simply done by adding an agent exhibiting the desired behaviour. Third, it minimizes the need for handcrafts. Fourth, can work with any action selection mechanism and integrates well with other external sources.

When we compare the results of the proposed architecture with those of the traditional architectures, we show the feasibility of the proposed architecture to bring an ability which was perceived challenging to achieve using traditional architectures, while maintaining a good performance score. We seek to determine the point of departure with these traditional architectures. It proved difficult to achieve logically advancing conversation using FSM [14], and Frame-based architectures [13] because, FSM architectures supported a fixed conversational path bounded within the states, also known as directional flow. Any deviation from this path lead to unexpected behaviour, unfortunately, natural conversation does not follow predefined paths. Frame-based, on the other hand use slot filling, which is limited to the information available in the slot. This meant only conversation taking a given flow of direction was permitted; just like the former, this goes against the idea of natural conversation.

While additional behaviour was supported through handcraft techniques, creating handcrafts to override the basic behaviour of architecture proved a complex task, moreover having many handcrafts in a system complicated its architecture. Previous studies have shown success of this traditional architectures in specific areas e.g. handling routine tasks such as in air ticket booking, it remains unclear how to quantify the individual contribution of such handcrafts. Besides, the degree such handcrafts can push the conversation, has not been confirmed. However, what is certain, is that handcrafts present the following challenges; (1) complicates the overall architecture (2) cannot be ported since their design was specific to a particular focus within a particular setting. (3) no handcrafts to solve all problems. We speculate that this could be one of the primary reason as to why not every domain used this technology.

The presented AI Chatbot was able to logically handle the progression in the conversations, and included sub dialogs intended to elicit more information. This ability is naturally demanded in order for some newer domains to flourish. Especially where TODS were not serving before. So are we likely to practically experience more task-oriented DS than their chat oriented counterparts, in the near future?

V. CONCLUSION

Although widely accepted or used, some traditional architectures by themselves act as a bottleneck towards improving conversational capabilities of AI Chatbots [13, 14]. While the future demands revolutionizing information seeking from static single query at a time, to a progressive kind of search. We demonstrate the possibility of enhancing conversational capabilities of TODS also AI Chatbots, by adopting better architectures and methods. Thus making them serve even in newer domains they could not serve before.

Based on our experiments, we speculate that if the novel architecture is adopted and improved, it will provide one useful approach to introducing new but desired feature(s) in TODS. Further work will be to develop the prototype to full scale AI Chatbot.

VI. 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 (IJSRCSEIT), vol. 3, no. 5, 2018.

[2] K. Mugoye, H. Okoyo and S. McOyowo. " MAS architectural model for dialog systems with advancing conversations," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 3, Issue 5, Nov-December 2018.

[3] K. Mugoye, H. Okoyo and S. McOyowo. " Smart-bot Technology: Conversational Agents Role in Maternal Healthcare Support," IST-Africa 2019 Conference Nairobi, Kenya, May 2019, Jan-May 2019.

[4] "Build natural and rich conversational experiences," Google, [Online]. Available: https://dialogflow.com/ . [Accessed 29 April 2019].

[5] M. A. Walker, D. J. Litman, C. A. Kamm and A. Abella, "PARADISE: A Framework for Evaluating Spoken Dialogue Agents," in ACL/EACL 35th Annual Meeting of the Association for Computational Linguistics, San Francisco, 1997.

[6] R. V. Solingen, V. Basili, G. Caldiera and H. D. Rombach, "Goal Question Metric (GQM) Approach.," in Encyclopedia of Software Engineering, 2002.

[7] J. C. Carletta, "Assessing the reliability of subjective codings," in Computational Linguistics, 1996.

[8] S. Siegel and N. J. Castellan, Nonparametric Statistics for the Behavioral Sciences, 2 ed., New York: McGraw-Hill, 1988.

[9] H. J. Wilson, P. R. Daugherty and N. Morini-Bianzino, "Will AI Create as Many Jobs as it Eliminates? MIT Sloan Management Review".

[10] K. Morrissey and J. Kirakowski, "'Realness' in Chatbots: Establishing Quantifiable Criteria," in Human-Computer Interaction, Berlin Heidelberg, 2013.

[11] M. Eeuwen, "Mobile conversational commerce: messenger chatbots as the next interface between businesses and consumers," 2017.

[12] D. Cohen and I. Lane, "An oral exam for measuring a dialog system's capabilities," in Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[13] E. Barnard, A. Halberstadt, C. Kotelly and M. Phillips, "A Consistent Approach to Designing Spoken dialog Systems," in Proceedings of ASRU'99 Conference, Keystone, Colorado, 1999.

[14] M. F. McTear, Spoken Dialogue Technology, Verlag-London : Springer, 2004.

Cite this article as : K. Mugoye, Dr. H. O. Okoyo, Dr. S. O. Mc Oyowo, "Towards Logically Progressive Dialog for Future TODS to Serve in Complex Domains", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 5, pp. 184-192, September-October 2019. Available at doi : https://doi.org/10.32628/CSEIT19558
Journal URL : http://ijsrcseit.com/CSEIT19558