Proceedings Paper:

Jafari, R orcid.org/0000-0001-7298-2363, Razvarz, S and Gegov, A (2020) End-to-End Memory Networks: A Survey. In: Arai, K, Kapoor, S and Bhatia, R, (eds.) Advances in Intelligent Systems and Computing. SAI 2020 Computing Conference, 16-17 Jul 2020, Online. Springer Verlag , pp. 291-300. ISBN 978-3-030-52245-2

https://doi.org/10.1007/978-3-030-52246-9_20

End-to-End Memory Networks: A Survey

Raheleh Jafari[1], Sina Razvarz[2], and Alexander Gegov[3]

[1] School of design, University of Leeds, Leeds, LS2 9JT, UK
r.jafari@leeds.ac.uk
[2] Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico
srazvarz@yahoo.com
[3] School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO1 3HE, UK
alexander.gegov@port.ac.uk

Abstract. Constructing a dialog system which can speak naturally with a human is considered as a major challenge of artificial intelligence. End-to-end dialog system is taken to be a primary research topic in the area of conversational systems. Since an end-to-end dialog system is structured based on learning a dialog policy from transactional dialogs in a defined extent, therefore, useful datasets are required for evaluating the learning procedures.
In this paper, different deep learning techniques are applied to the Dialog bAbI datasets [1]. On this dataset, the performance of the proposed techniques is analyzed. The performance results demonstrate that all the proposed techniques attain decent precisions on the Dialog bAbI datasets. The best performance is obtained utilizing end-to-end memory network with a unified weight tying scheme (UN2N).

Keywords: memory networks, deep learning, Dialog bAbI dataset

1 Introduction

Instructing machines that can converse like a human for real-world objectives is possibly one of the crucial challenges in artificial intelligence. In order to construct a meaningful conversation with human, the dialog system is required to be qualified in the perception of natural language, constructing intelligent decisions as well as producing proper replies [2–4]. Dialog systems, recognized as interactive conversational agents, communicate with the human through natural language in order to aid, supply information and amuse. They are utilized in an extensive applications domain from technical support services to language learning tools [5, 6].

Artificial intelligence techniques are viewed as the most efficient techniques in recent decades [7–18]. For example, Fuzzy logic systems are broadly utilized to model the systems characterizing vague and unreliable information [19–38]. In artificial intelligence area [39, 40], end-to-end dialog systems have been attained interest because of the current progress of deep neural networks. In [41] a gated end-to-end trainable memory network is proposed which is learning in an end-to-end procedure without the utilization of any extra supervision signal. In [1] the original task is broken down into short tasks where they should be individually learned by the agent, and also built in order to perform the original task. In [42] a long short term memory (LSTM) model is suggested which learns in order to interact with APIs on behalf of the user. In [43] a dynamic memory network is introduced which contains tasks for part-of-speech classification as well as question answering, also uses two gated recurrent units in order to carry out inference. In [44] the memory network has been implemented which needed supervision in every layer of the network. In [45] a set of four tasks in order to test the capability of end-to-end dialog systems has been introduced which focuses on the domain of movies entities. In [46] a word-based method to dialog state tracking utilizing recurrent neural networks (RNNs) is proposed which needs less feature engineering. Even though neural network models include a tiny learning pipeline, they need a remarkable content of the training. Gated recurrent network (GRU) and LSTM units permit RNNs to deal with the longer texts needed for question answering. Additional advancements to be mentioned as attention mechanisms, as well as memory networks, permit the network to center around the most related facts.

In this paper, the applications of different types of memory networks are studied on data from the Dialog bAbI. The performance results demonstrate that all the proposed techniques attain decent precisions on the Dialog bAbI datasets. The best performance is obtained utilizing UN2N. The remaining of the article is organized as follows. In Section 2, different types of memory networks are demonstrated and explained in details. Experimental results are given in Section 3. Section 4 concludes the work.

2 Memory Networks

2.1 End-to-End Memory Network with Single Hop

The end-to-end Memory Network (N2N) with single hop has two stories embedding $\widetilde{A}$, $\widetilde{C}$, as well as a question embedding $\widetilde{B}$, see Figure 1. Matrices dot product are utilized in order to match each word in the story with each word in the question which will cause the creation of the attention. By passing the attention through a softmax layer they will change into the probability distribution across the whole word from the story. Afterward, these probabilities are implemented to the story embedding $\widetilde{C}$ and the sum of that with the question embedding $\widetilde{B}$ passes through a dense layer and the softmax prediction layer.

2.2 End-to-End Memory Network with Stacked Hops

The N2N architecture contains two major components: supporting memories and final answer prediction [47]. Supporting memories consist of a set of input and output memory represented by memory cells. In complicated tasks with

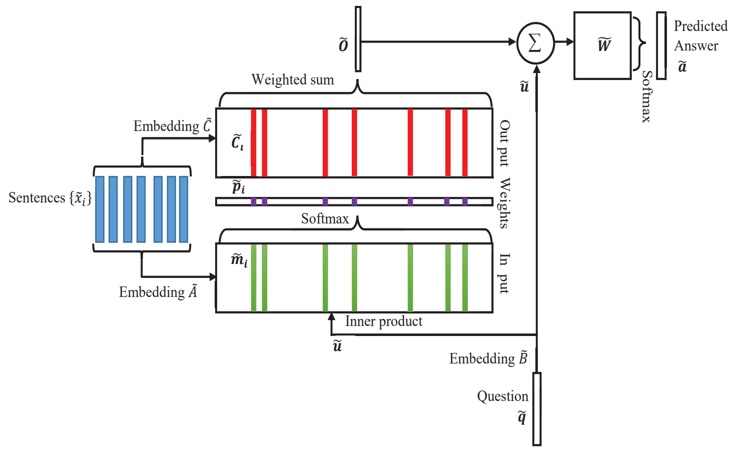

Fig. 1. End-to-end memory network with single hop

the requirement of multiple supporting memories, the model can be developed in order to contain more than one set of input-output memories by stacking a number of memory layers. Each memory layer in the model is called hop, also the input of the $(\kappa+1)^{th}$ hop is the output of the κ^{th} hop:

$$\tilde{u}^{\kappa+1} = \tilde{o}^{\kappa} + \tilde{u}^{\kappa} \tag{1}$$

Each layer contains its own embedding matrices $\widetilde{A}^{\kappa}, \widetilde{C}^{\kappa}$, utilized in order to embed the inputs $\tilde{x}_i$.

The prediction of the answer to the question $\tilde{q}$, is carried out by

$$\tilde{a} = softmax(\widetilde{W}(\tilde{o}^{\kappa} + \tilde{u}^{\kappa})) \tag{2}$$

where $\tilde{a}$ is taken to be the predicted answer distribution, $\widetilde{W}$ (of size $V \times d$) is considered to be a parameter matrix for the model in order to learn, also κ is the total number of hops.

The N2N architecture with three hop operations is shown in Figure 2. The hard max operations within each layer are substituted with a continuous weighting from the softmax.

The method takes a discrete set of inputs $\tilde{x}_1, ..., \tilde{x}_n$ which are stored in the memory, a question $\tilde{q}$, also outputs a reply $\tilde{a}$. The model can write all $\tilde{x}$ to the memory up to a fixed buffer size, also it obtains a continuous demonstration for $\tilde{x}$ and $\tilde{q}$. Afterward, the continuous demonstration is processed with multiple hops in order to generate $\tilde{a}$. This permits backpropagation of the error signal through multiple memory accesses back to the input while training.

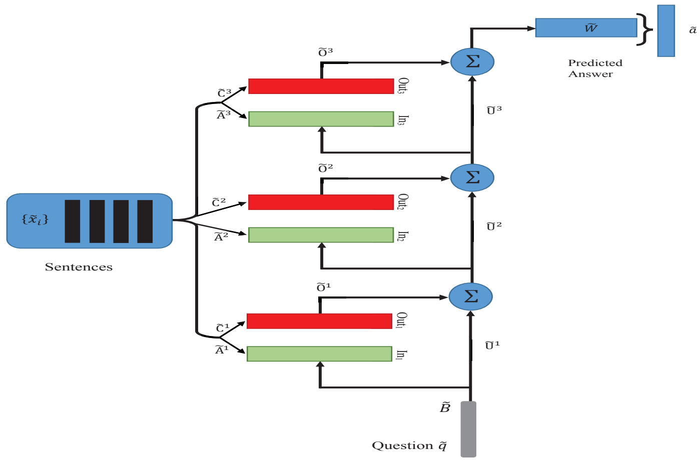

Fig. 2. A three layer end-to-end memory network

2.3 Gated End-to-End Memory Network

The gated end-to-end memory network (GN2N) is able to dynamically conditioning the memory reading operation on the controller state $\tilde{u}^{\kappa}$ at every hop, see Figure 3. In GN2N, (1) is reformulated as below [48],

$$T^{\kappa}(\tilde{u}^{\kappa}) = \sigma(\widetilde{W}_T^{\kappa}\tilde{u}^{\kappa} + \tilde{b}_T^{\kappa}) \tag{3}$$

$$\tilde{u}^{\kappa+1} = \tilde{o}^{\kappa} \odot T^{\kappa}(\tilde{u}^{\kappa}) + \tilde{u}^{\kappa} \odot (1 - T^{\kappa}(\tilde{u}^{\kappa})) \tag{4}$$

where $\widetilde{W}_T^{\kappa}$ and $\tilde{b}^{\kappa}$ are taken to be the hop-specific parameter matrix and bias term for the κ^{th} hop respectively. $T^{\kappa}(\tilde{x})$ is the transform gate for the κ^{th} hop. $\odot$ is the Hadamard product.

2.4 End-to-End Memory Networks with Unified Weight Tying

In [47], two kinds of weight tying are proposed for N2N, namely adjacent and layer-wise. Layer-wise approach portions the input and output embedding matrices across various hops ($i.e.$, $\widetilde{A}^1 = \widetilde{A}^2 = ... = \widetilde{A}^{\kappa}$ and $\widetilde{C}^1 = \widetilde{C}^2 = ... = \widetilde{C}^{\kappa}$). Adjacent approach portions the output embedding for a given layer with the corresponding input embedding ($i.e.$, $\widetilde{A}^{\kappa+1} = \widetilde{C}^{\kappa}$). Furthermore, the matrix $\widetilde{W}$ which predicts the answer, as well as the question embedding matrix $\widetilde{B}$, are developed as $\widetilde{W}^T = \widetilde{C}^{\kappa}$ and $\widetilde{B} = \widetilde{A}^1$. In [48], a dynamic mechanism is designed which permits the model to choose the proper kind of weight tying on the basis of the input. Therefore, the embedding matrices are developed dynamically for every instance which makes UN2N more efficient compared with N2N and GN2N where the same embedding matrices are implemented for each input. In UN2N

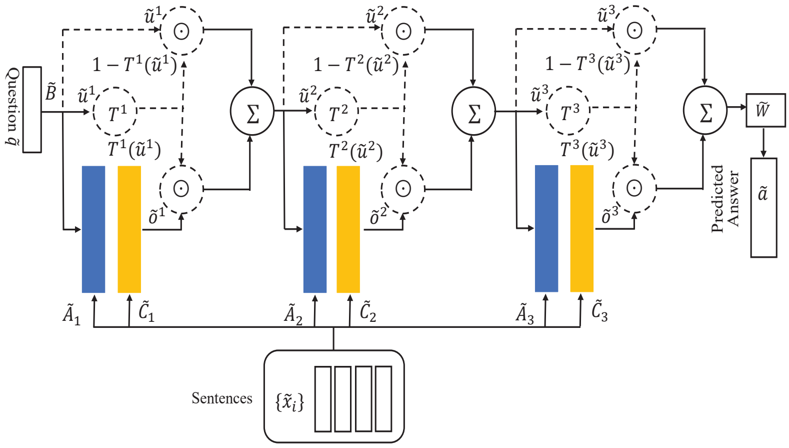

Fig. 3. Gated end-to-end memory network

a gating vector $\tilde{z}$, described in (8), is used in order to develop the embedding matrices, $\widetilde{A}^{\kappa}$, $\widetilde{C}^{\kappa}$, $\widetilde{B}$, and $\widetilde{W}$. The embedding matrices are influenced by the information transported by $\tilde{z}$ related to the input question $\tilde{u}^0$ and the context sentences in the story $\tilde{m}_t$. Therefore,

$$\widetilde{A}^{\kappa+1} = \widetilde{A}^{\kappa} \odot \tilde{z} + \widetilde{C}^{\kappa} \odot (1 - \tilde{z}) \tag{5}$$

$$\widetilde{C}^{\kappa+1} = \widetilde{C}^{\kappa} \odot \tilde{z} + \widetilde{C}^{\kappa+1} \odot (1 - \tilde{z}) \tag{6}$$

where $\odot$ is taken to be the column element-wise multiplication operation, also $\widetilde{C}^{\kappa+1}$ is the unconstrained embedding matrix. In (5) and (6), the large value of $\tilde{z}$ leads UN2N towards the layer-wise approach and the small value of $\tilde{z}$ leads UN2N towards the adjacent approach.

In UN2N, at first, the story is encoded by reading the memory one step at a time with a gated recurrent unit (GRU) as below,

$$\tilde{h}_{t+1} = GRU(\tilde{m}_t, \tilde{h}_t) \tag{7}$$

such that t is considered to be the recurrent time step, also $\tilde{m}_t$ is taken to be the context sentence in the story at time t. Afterward, the following relation is defined,

$$\tilde{z} = \sigma(\widetilde{W}_{\tilde{z}} \begin{bmatrix} \tilde{u}^0 \\ \tilde{h}_T \end{bmatrix} + \tilde{b}_{\tilde{z}}) \tag{8}$$

where $\tilde{h}_T$ is the last hidden state of the GRU which presents the story, $\widetilde{W}_{\tilde{z}}$ is considered as a weight matrix, $\tilde{b}_{\tilde{z}}$ is bias, σ is taken to be the sigmoid function also, $\begin{bmatrix} \tilde{u}^0 \\ \tilde{h}_T \end{bmatrix}$ is the concatenation of $\tilde{u}^0$ and $\tilde{h}_T$. A linear mapping $G \in R^{d \times d}$ is added for updating the connection between memory hops as below,

$$\tilde{u}^{\kappa+1} = \tilde{o}^{\kappa} + (G \odot (1 - \tilde{z}))\tilde{u}^{\kappa} \tag{9}$$

3 Experiments and Results

3.1 Experiment Setup

In this section, an extensive range of parameter settings along with data set configurations are utilized in order to validate the proposed techniques in this paper.

3.2 Task explanations

The tasks in the dataset are divided into 5 groups where each group focus on a special objective.

Task 1: Issuing API calls The chatbot asks questions in order to fill the missing areas, and finally produces a valid corresponding API call. The questions asked by the bot is for collecting information in order to make the prediction possible.

Task 2: Updating API calls In this part users update their requests. The chatbot asks from users if they have finished their updates, then chatbot generates updated API call.

Task 3: Demonstrating options The chatbot provides options to users utilizing the corresponding API call.

Task 4: Generating additional information User can ask for the phone number and address and the bot should use the knowledge bases facts correctly in order to reply.

Task 5: Organizing entire dialogs Tasks 1-4 are combined in order to generate entire dialogs.

For evaluating the capability of the techniques in order to deal with out-of-vocabulary (OOV) items a set of test data is used which contains entities different from the training set. Task 6 is the Dialog state tracking 2 task (DSTC-2) [49] with real dialogs, and only has one setup.

3.3 Experimental Results

Efficiency results on Dialog bAbI tasks are demonstrated in Table 1, with seven techniques which are among the most important techniques, namely rule-based systems, TF-IDF, nearest neighbor, supervised embedding, N2N, GN2N, and UN2N. As is shown in Table 1, the rule-based system has a high performance on tasks 1-5. However, its performance reduces when dealing with DSTC-2 task. TF-IDF match has poor performance compared with other methods on both the

Table 1. The accuracy results of rule-based systems, TF-IDF, nearest neighbor, supervised embedding, N2N, GN2N, and UN2N methods

Task	Rule-based Systems	TF-IDF Match		Nearest Neighbor	-match				match			
		no type	type		S-Emb	N2N	GN2N	UN2N	S-Emb	N2N	GN2N	UN2N
1. Issuing API calls	100.0	5.6	22.4	55.1	100.0	99.9	100.0	100.0	83.2	100.0	100.0	100.0
2. Updating API calls	100.0	3.4	16.4	68.3	68.4	100.0	100.0	100.0	68.4	98.3	100.0	100.0
3. Displaying options	100.0	8.0	8.0	58.8	64.9	74.9	74.9	74.9	64.9	74.9	74.9	74.9
4. Generating additional information	100.0	9.5	17.8	28.6	57.2	59.5	57.2	57.2	57.2	100.0	100.0	100.0
5. Organizing entire dialogs	100.0	4.6	8.1	57.1	75.4	96.1	96.3	99.2	76.2	93.4	98.0	99.4
Average	**100.0**	**6.2**	**14.5**	**53.6**	**73.2**	**86.1**	**85.7**	**86.3**	**70.0**	**93.3**	**94.6**	**99.4**
1. (OOV) Issuing API calls	100.0	5.8	22.4	44.1	60.0	72.3	82.4	83.0	67.2	96.5	100.0	100.0
2. (OOV) Updating API calls	100.0	3.5	16.8	68.3	68.3	78.9	78.9	78.9	68.3	94.5	94.2	94.5
3. (OOV) Displaying options	100.0	8.3	8.3	58.8	65.0	74.4	75.3	75.2	65.0	75.2	75.1	75.3
4. (OOV) Generating additional information	100.0	8.8	17.2	28.6	57.0	57.6	57.0	57.0	57.1	100.0	100.0	100.0
5.(OOV) Organizing entire dialogs	100.0	4.6	9.0	48.4	58.2	65.5	66.7	67.8	64.4	77.7	79.4	79.5
Average	**100.0**	**6.4**	**14.7**	**49.6**	**61.7**	**69.7**	**72.1**	**72.4**	**64.4**	**88.8**	**89.7**	**89.8**
6. Dialog state tracking 2	33.3	1.6	1.6	21.9	22.6	41.1	47.4	42.4	22.1	41.0	48.7	42.9

simulated tasks 1-5 and on the real data of task 6. The performance of the TF-IDF match with match type features considerably increases but is still behind the nearest neighbor technique. Supervised embedding has higher performance compared with TF-IDF match and nearest neighbor technique. In task 1, supervised embedding is fully successful but its performance reduces in task 2-5, even with match type features. GN2N and UN2N models outperform the other methods in DSTC-2 task and Dialog bAbI tasks respectively.

4 Conclusion

End-to-end learning scheme is suitable for constructing the dialog system because of its simplicity in training as well as effectiveness in model updating. In this paper, the applications of various memory networks are studied on data from the Dialog bAbI. The performance results demonstrate that all the proposed techniques attain decent precisions on the Dialog bAbI datasets. The best performance is obtained utilizing UN2N. In order to evaluate the true performance of the proposed methods, extra experimentations are required utilizing wide non-synthetic data set.

References

1. Bordes, A., Weston, J.: Learning end-to-end goal-oriented dialog, *arXiv preprint arXiv:1605.07683,* (2016).
2. Araujo, T.: Living up to the chatbot hype: The influence of anthropomorphic design cues and communicative agency framing on conversational agent and company perceptions, *Computers in Human Behavior,* 85, 183-189, (2018).

3. Hill, J., Ford, W.R., Farreras, I.G.: Real conversations with artificial intelligence: A comparison between humanhuman online conversations and human-chatbot conversations, *Computers in Human Behavior,* 49, 245-250, (2015).
4. Quarteroni, S.: A Chatbot-based Interactive Question Answering System, *11th Workshop on the Semantics and Pragmatics of Dialogue,* 83-90, (2007).
5. Young, S., Gasic, M., Thomson, B., Williams, J.D.: POMDP-based statistical spoken dialog systems: A review, *PROC IEEE,* 101, 1160-1179, 2013.
6. Shawar, B.A., Atwell, E.: Chatbots: are they really useful?, *LDV Forum,* 22, 29-49, (2007).
7. Dote, Y., Hoft, R.G.: Intelligent ControlPower Electronics Systems, *Oxford, U.K.: Oxford Univ. Press,* (1998).
8. Mohanty, S.: Estimation of vapour liquid equilibria for the system carbon dioxidedifluoromethane using artificial neural networks, *International Journal of Refrigeration,* 29, 243249, (2006).
9. Razvarz, S., Jafari, R., Yu, W., Khalili, A.: PSO and NN Modeling for Photocatalytic Removal of Pollution in Wastewater, *14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) Electrical Engineering,* 1-6, (2017).
10. Jafari, R., Yu, W.: Artificial neural network approach for solving strongly degenerate parabolic and burgers-fisher equations, *12th International Conference on Electrical Engineering, Computing Science and Automatic Control,* doi:10.1109/ICEEE.2015.7357914, (2015).
11. Jafari, R., Razvarz, S., Gegov, A.: A New Computational Method for Solving Fully Fuzzy Nonlinear Systems, *In: Computational Collective Intelligence. ICCCI 2018. Lecture Notes in Computer Science, Springer, Cham,* 11055, 503-512, (2018).
12. Razvarz, S., Jafari, R.: ICA and ANN Modeling for Photocatalytic Removal of Pollution inWastewater, *Mathematical and Computational Applications,* 22, 38-48, (2017).
13. Razvarz, S., Jafari, R., Gegov, A., Yu, W., Paul, S.: Neural network approach to solving fully fuzzy nonlinear systems, *Fuzzy modeling and control Methods Application and Research, Nova science publisher, Inc, NewYork.* ISBN: 978-1-53613-415-5, 45-68 (2018).
14. Razvarz, S., Jafari, R.: Intelligent Techniques for Photocatalytic Removal of Pollution in Wastewater, *Journal of Electrical Engineering,* 5, 321-328, doi: 10.17265/2328-2223/2017.06.004, (2017).
15. Graupe, D.: Chapter 112. In: Chen W, Mlynski DA (eds) Principles of artificial neural networks, *Advanced Series in circuits and systems,* 3, 1st edn. World Scientific, p. 4e189, (1997).
16. Jafari, R., Yu, W., Li, X.: Solving Fuzzy Differential Equation with Bernstein Neural Networks, *IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary,* 1245-1250 (2016).
17. Jafari, R. Yu, W.: Uncertain nonlinear system control with fuzzy differential equations and Z-numbers, *18th IEEE International Conference on Industrial Technology, Canada,* 890-895, doi:10.1109/ICIT.2017.7915477, (2017).
18. Jafarian, A., Measoomy nia, S., Jafari, R.: Solving Fuzzy Equations Using Neural Nets with a New Learning Algorithm, *Journal of Advances in Computer Research,* 3, 33-45, (2012).
19. Werbos, P.J.: Neuro-control and elastic fuzzy logic: Capabilities, concepts, and applications, *IEEE Trans. Ind. Electron.,* 40, 170180, (1993).

20. Jafari, R., Yu, W., Razvarz, S., Gegov, A.: Numerical methods for solving fuzzy equations: A Survey, *Fuzzy Sets and Systems,* ISSN 0165-0114, https://doi.org/10.1016/j.fss.2019.11.003, (2019).
21. Kim, J.H., Kim, K.S., Sim, M.S., Han, K.H., Ko, B.S., An application of fuzzy logic to control the refrigerant distribution for the multi type air conditioner, *in Proc. IEEE Int. Fuzzy Systems Conf.,* 3, 1350-1354, (1999).
22. Wakami, N., Araki, S., Nomura, H.: Recent applications of fuzzy logic to home appliances, *in Proc. IEEE Int. Conf. Industrial Electronics, Control, and Instrumentation, Maui, HI,* 155160, (1993).
23. Jafari, R., Razvarz, S.: Solution of Fuzzy Differential Equations using Fuzzy Sumudu Transforms, *IEEE International Conference on Innovations in Intelligent Systems and Applications,* 84-89, (2017).
24. Jafari, R., Razvarz, S., Gegov, A., Paul, S.: Fuzzy modeling for uncertain nonlinear systems using fuzzy equations and Z-numbers, *Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK . Advances in Intelligent Systems and Computing, Springer,* 840, 66-107 (2018).
25. Jafari, R., Razvarz, S.: Solution of fuzzy differential equations using fuzzy sumudu transforms, *Mathematical and Computational Applications,* 1-15, (2018).
26. Jafari R., Razvarz S., Gegov A.: Solving Differential Equations with Z-Numbers by Utilizing Fuzzy Sumudu Transform, *Intelligent Systems and Applications. IntelliSys 2018. Advances in Intelligent Systems and Computing, Springer, Cham.* 869, 1125-1138, (2019).
27. Yu, W., Jafari, R.: Fuzzy Modeling and Control with Fuzzy Equations and Z-Number, *IEEE Press Series on Systems Science and Engineering, Wiley-IEEE Press,* ISBN-13: 978-1119491552, (2019).
28. Negoita, C.V., Ralescu, D.A.: Applications of Fuzzy Sets to Systems Analysis, *Wiley, New York,* (1975).
29. Zadeh, L.A.: Probability measures of fuzzy events, *Journal of Mathematical Analysis and Applications,* 23, 421-427, (1968).
30. Zadeh, L.A.: Calculus of fuzzy restrictions, *Fuzzy sets and Their Applications to CognitiveandDecision Processes, Academic Press, New York,* 1-39, 1975.
31. Zadeh, L.A.: Fuzzy logic and the calculi of fuzzy rules and fuzzy graphs, *Multiple-Valued Logic,* 1, 1-38, (1996).
32. Razvarz, S., Jafari, R.: Experimental study of Al2O3 nanofluids on the thermal efficiency of curved heat pipe at different tilt angle, *Journal of Nanomaterials,* 1-7, (2018).
33. Razvarz, S., Vargas-Jarillo, C., Jafari, R.: Pipeline Monitoring Architecture Based on Observability and Controllability Analysis, *IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany,* 1, 420-423, doi:10.1109/ICMECH.2019.872287, (2019).
34. Razvarz, S., Vargas-jarillo, C., Jafari, R., Gegov, A.: Flow Control of Fluid in Pipelines Using PID Controller, *IEEE Access,* 7, 25673-25680, doi:10.1109/ACCESS.2019.2897992, (2019).
35. Razvarz, S., Jafari, R.: Experimental Study of Al2O3 Nanofluids on the Thermal Efficiency of Curved Heat Pipe at Different Tilt Angle, *2nd international congress on technology engineering and science (ICONTES),* Malaysia, (2016).
36. Jafari, R., Razvarz, S., Gegov, A.: Neural Network Approach to Solving Fuzzy Nonlinear Equations using Z-Numbers, *IEEE Transactions on Fuzzy Systems,* doi: 10.1109/TFUZZ.2019.2940919, (2019).

37. Jafari, R., Yu, W., Li, X., Razvarz, S.: Numerical solution of fuzzy differential equations with Z-numbers using bernstein neural networks, *International Journal of Computational Intelligence Systems,* 10, 1226-1237, (2017).
38. Jafari, R., Yu, W., Li, X.: Numerical solution of fuzzy equations with Z-numbers using neural networks, *Intelligent automation and Soft Computing,* 1-7, (2017).
39. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks, *Proc. of ICML-2011,* (2011).
40. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and Tell: A neural image caption generator, *Proc. of CVPR-2015,* (2015).
41. Liu, F., Perez, J.: Gated end-to-end memory networks, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Association for Computational Linguistics, Valencia, Spain,* 1-10, (2017).
42. Williams, J.D., Zweig, G.: End-to-end lstm-based dialog control optimized with supervised and reinforcement learning, *arXiv preprint arXiv:1606.01269,* (2016).
43. Kumar, A., Irsoy, O., Su, J., Bradbury, J., English, R., Pierce, B., Ondruska, P., Gulrajani, I., Socher, R.: Ask me anything: Dynamic memory networks for natural language processing, *Proc. of ICML-2016,* (2016).
44. Weston, J., Chopra, S., Bordes, A.: Memory networks, *International Conference on Learning Representations (ICLR),* (2015).
45. Dodge, J., Gane, A., Zhang, X., Bordes, A., Chopra, S., Miller, A.H., Szlam, A., Weston, J.: Evaluating prerequisite qualities for learning end-to-end dialog systems, *Proc. of ICLR-2016,* (2016).
46. Henderson, M., Thomson, B., Young, S.: Word-based dialog state tracking with recurrent neural networks, *Proc. of SIGDIAL-2014,* (2014).
47. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks, *Proceedings of Advances in Neural Information Processing Systems (NIPS 2015), Montreal, Canada,* 2440-2448, (2015).
48. Liu, F., Cohn, T., Baldwin, T.: Improving End-to-End Memory Networks with Unified Weight Tying, *Proceedings of the 15th Annual Workshop of The Australasian Language Technology Association (ALTW 2017), Brisbane, Australia,* 16-24, (2017).
49. Henderson, M., Thomson, B., Williams, J.D.: The second dialog state tracking challenge, *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL 2014), Philadelphia, USA,* 263-272, (2014).