International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 1885–1895

Framework for Story Generation using RNN

Pallavi Vanarse[1], Sonali Jadhav[2], Snehashri Todkar[3], Prof. Kunjali R. Pawar[4]

[1]Department of Information Technology Government College of Engineering, Karad, India,
[2]Department of Information Technology Government College of Engineering, Karad, India,
[3]Department of Information Technology Government College of Engineering, Karad, India,
[4]Assistant Professor, Department of Information Technology Government College of Engineering, Karad, India

Abstract

In story generation the problem that occurs mostly is selection of events that satisfies the criteria of story creation. It is knowledge-intensive. Conventional story generation frameworks depend on prior well-defined domain paradigms about fictional-world, including characters, scenes and events which should be included. In this paper, we prospect a public-domain story generation framework that constructs stories while input given is storyline. A framework is proposed that first frames a story based on storyline with help of hierarchical neural networks. A technique is suggested for automated generation of story through which the framework decomposes a storyline into the generation of sequential events and the generation of spoken-written language sentences from events. The algorithm is used to extract stories from text and dataset further to compute optimal story. The framework produces an example of computational artistry of resourcefulness that can be comparable against generous writing.

Keywords: *Deep Learning, Recurrent Neural networks, Story generation, Storyline, LSTM.*

1. Introduction

Generally, computer systems in AI have been created either with the objective of producing programs that perform a tasks similar to human beings, or with the objective of producing programs that perform a task in any way that satisfies objective criteria of validity and efficiency, irrespective of how human beings do it[9]. In this paper we introduce a model that minimizes the gap between these two objectives: A framework automated generation of story through which decompose a storyline into the generation of logically connected events. The techniques used for designing a framework for spoken-written language generation have been explored to many paradigms but in related text generation tasks.

A narration of event or story is text-format which is a logically connected set of sequences of events which involves some shared characters and places. The main problem in creation of stories using RNN (Recurrent Neural Network) is automatically select a sequence of events, actions, or words that can be pass down to tell as s story. For describing an understandable arrangement of event it is necessary to compose the natural language texts. Symbolic planning or Case-based reasoning is used by many story generation models [8]. The main factors which are considered during writing a story are: it must be concise and it must allow the reader to explore different areas of story which are of interest of him/her. While these automated story generation systems were able to accomplish significant results, these systems depends on a expertise engineer to provide symbolic domain models that defined acknowledged characters, actions, and knowledge about when character actions can and cannot be performed [8].

The assumption is that storytelling systems can be get acquisition from storyline construction to generate more rational, well-organized and on-topic stories [1].Human and computer can communicate with each other and works together to enable much more potentially entertaining interactions. This is an additional advantage of this plan-and- writes schema.

The Section 2 represents literature review, section 3 represents the design methodology it contains problem formulation, architecture and working of framework . Algorithm is evaluated in Section 4.Section 5 is most important which contains expected results in which accurate result is generated and last but not least section 6 contains conclusion and future scope.

2. Literature Review

As artificial intelligence comes to existence, automated story generation has become one of the important researches of interest. In automated story generation the problem is selection of sequence of events or actions which are used for creating a story. Most of the story generating models requires a domain model that contains all information about the characters and places of respective domain. Further the model creates some action which can be performed by that character. RNN are extremely powerful for NLP tasks having successful results on tasks like speech recognition and machine translation. Previous work on story generation has explored seq2seq RNN architecture [7, 11, and 14]. For an example using photos to create short paragraphs which describe the photo, generating technique that summarizes sequences in a movie. LSTM [11, 12, and 15] works as it learns a word, then sentence then paragraph embedding and then forms the paragraphs into a text. RNN encodes the language models more accurately than previously used statistics. It can be said that it learns to predict the probability of next character, word or sentence in a story [2].

[1]Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, Rui Yan1, "Plan-and-Write: Towards Better Automatic Storytelling"

The proposed plan-and-write framework that generate stories from given titles and leverages storylines to improve the diversity and coherence of the generated story. Two strategies: dynamic and static planning are explored and compared. Authors develop evaluation metrics to measure the diversity of the generated stories, and conduct novel analysis to examine the importance of different aspects of stories for human evaluation.
The dataset used to conduct the experiments on the ROC-Stories corpus. It contains 98,162 short commonsense stories as training data, and additional 1,817 stories for development and test, respectively. The stories are five-sentence stories that capture a rich set of causal and temporal commonsense relations between daily events, making them a good resource for training storytelling models.

[2]Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha and Karthik Sankaranarayanan, "Story Generation from Sequence of Independent Short Descriptions"

The proposed system of story generation is framed as a sequence-to-sequence neural machine learning network that learns latent representations of the input descriptions to generate output summaries.Authors used the MOSES toolkit. For phrase based translation model learning,grow-diag-final and heuristic is considered and to tackle lexicalized reordering, msd bidirectional-fe model.
The dataset used for experiments 'Text Annotations' from recently introduced Visual Storytelling Dataset (VIST).'Descriptions of Images-in-Isolation (DII)' and 'Stories of Images-in-Sequence (SIS)' are used.There are 41300 image sequences aligned with caption and story pairs.

[3] Tianming Wang, Xiaojun Wan, "Hierarchical Attention Networks for Sentence Ordering,"

A hierarchical structure made up of a word encoder and a sentence encoder-decoder. The word encoder learns the sentence embedding and the sentence encoder-decoder adjusts representations of sentences in context and makes arrangement of these sentences into a coherent text. Sentence decoder and sentence decoder are also composed of a stack of M attention layers. These attention layers have sub-layers. For the evaluation of the proposed model three datasets are used, the arXiv dataset (Chen, Qiu, and Huang

2016), the VIST dataset (Huang et al. 2016) and the ROC-Story dataset. Different methods are used among which LSTM+Set2Seq had the best performance.

[4]Melissa Roemmele, "Writing Stories with Help from Recurrent Neural Networks"

The author built an application called 'Creative Help' that provides automated writing assistance (Roemmele and Gordon 2015). In this application, a user writes a story and when she types \help\, the application returns a suggested next sentence which the user can edit like any other text in the story. The approach to generation in Creative Help used information retrieval methods to find stories similar to the user's story among a large corpus, and then extract sentences from these stories as suggested continuations.

Our approach builds off recent work in story generation. This thesis helps to construct the stories automatically to implement an RNN-based system. In this system, the user types a story-line, and then model returns a story. This functionality addresses an existing weakness in language generation research [3]. The goal of this approach is to make this problem traceable: by using words for sampling and implementing vocabulary constraints. This paper advances earlier work [1] toward learning models that used for story generation. This model presents both qualitative as well as quantitative analysis of different approaches in this domain.

3. Design Methodology

A. Problem Definition

Input: A storyline $t = \{t_1, t_2, \ldots.., t_n\}$is given to the framework to constrain writing, where t_i is the i^{th} word in a story-line.

Output: The system generates a story $p = \{p_1, p_{2,\ldots..,}p_m\}$ based on a title, where p_i denotes a sentence in the story

B. Architecture and Working of framework

The focus is to adopt recurrent neural network to contrivance our story generation framework, as they become very effective in implementing machine translation and image captioning frameworks. The workflow of our framework is presented in Fig. 1.

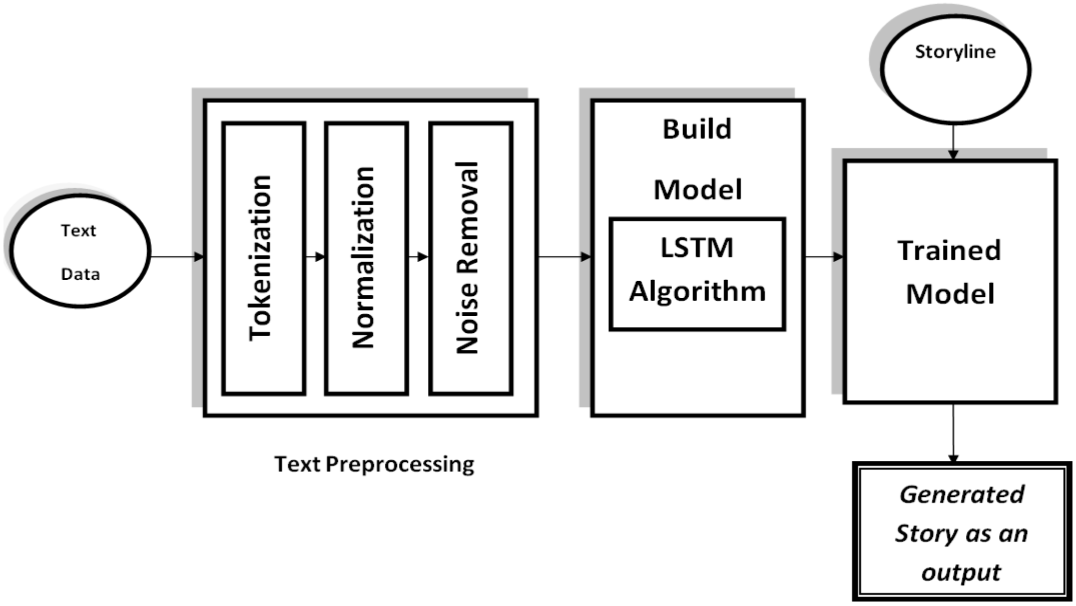

Fig.1. Architecture of story generation framework

The framework uses Dynamic approach as it gives flexibility. Firstly, the text file is given as an input to the framework then this text file undergoes preprocessing as the text preprocessing is essential footstep for Natural language processing that denatures the text into more simplex format. In Text preprocessing there are three main components:

A. Tokenization:

It is the process of dividing text document into sentences and sentences into words, that is, into small chunks .These small chunks are called as tokens. E.g. “Sky is blue.”, This process separates the sentences into separate words as “Sky”, “is” and “blue”. We can split them again into character as well.

B. Normalization:

It converts entire characters in file to the lowercase e.g. “Play” to “play”. If we don’t convert them into lowercase then they both refer as two different words and result in increasing dimensions so, converting it to lowercase reduces the vocabulary that framework must learn.

C. Noise Removal:

It removes the extra whitespaces from the text file. It is also known as Text cleaning.

D. Removing Stopwords:

The words like “a”,”the”,”to” are occurs repeatedly in text file. It is useful to get specific and important information from the given text.

E. Stemming and lemmatization:

Lemmatization is the way of shorten the words to their root stems. It reduces a word to its root word. Stemming can be done by erasing the end and beginning i.e. affixes and suffixes of root words. Lemmatization is more preferable.

E.g. Caring-> care ==> lemmatization
Caring ->car ==> stemming

F. Mapping words with integers:

It first creates the group of all distinct words and then mapping of each word to the integer.

4. Algorithm

The algorithm used by us is Long short-term memory (LSTM).It is special type of recurrent neural network (RNN), having capability of learning long term dependencies [1]. LSTM is used to remember things for short or long duration of time. All recurrent neural networks contain sequence of repeating units of neural network. In typical RNNs, this repeating unit will have a very basic structure like, single activation function [2]. The most basic RNNs are ambitious to train due to vanishing and exploding gradient problem.

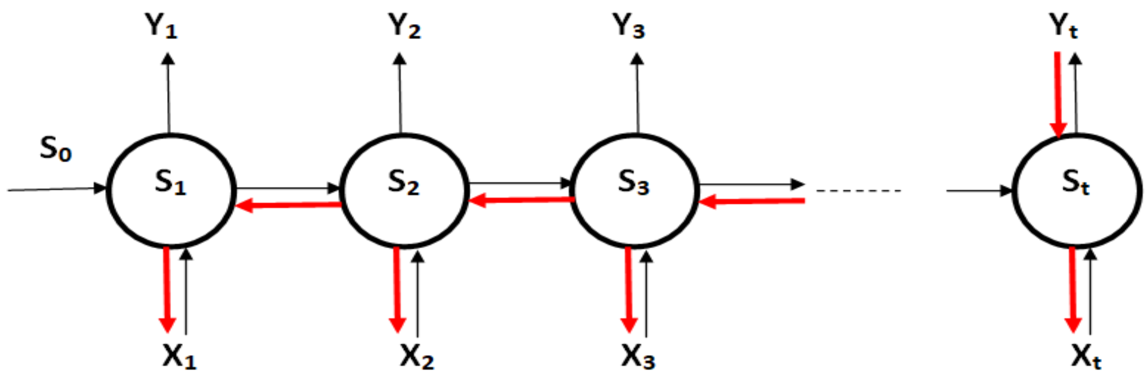

Fig.2. Prediction Process

Fig. 2 illustrate that in order to understand what would be the coming word in the sequence , the Recurrent neural network must memories the preceding context and then anticipate the next meaningful word.

Besides of having single neural network layer, In LSTM there are four connecting layers communicating in a special way.

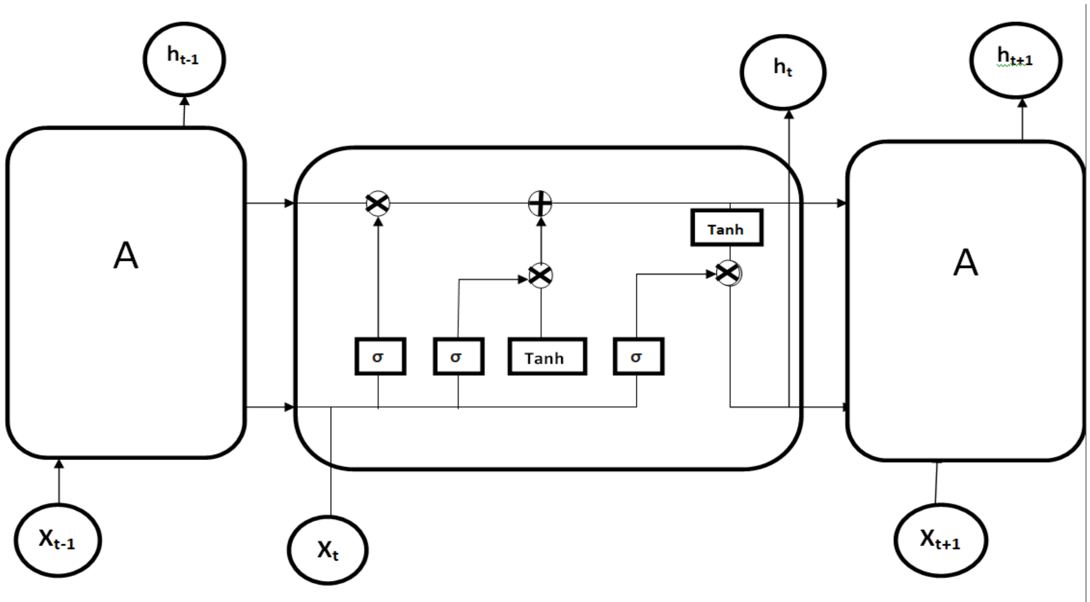

Fig.3. LSTM working

Step 1: Determines how much of the part it should recall.

First step in LSTM is to resolve which information to be excluded from the cell in that particular time frame. It is determined by sigmoid function. It take a look at the previous state (ht-1) and present input x_t and figure out the function.

$$F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

Where f_t is forget gate decides which information is to be emitted that is not important from previous time frame.

Step 2: Determines how much should this unit add to the present state.

In the second step, there are two parts. One is the sigmoid function and other is tanh activation function.Sigmoid function also known as Squashing function which gives the output to a range between 0 and 1.Tanh function gives the weightage to the values which are passed deciding their levels of importance (-1 to 1).

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$C_t = \tanh (W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

Where i_t is input gate determines which information pass through based on its significance in the current time frame.

Step 3: Determine what part of the present cell state continue to the output.

The third step is to determine what will be our output. First run the sigmoid function which decides what parts of the cell state gives the output then, pass the cell state through tanh to push the values to be between -1 and 1 and take a product of both[7].

$$o_t = \sigma (W_o \cdot [h_{t-1}, x_t] + b_o) \quad (4)$$

$$h_t = o_t * \tanh (C_t) \quad (5)$$

Where o_t is output gate allows the passed in information to impact the output in the present time frame. After Modeling of entire training dataset is used to learn the possibility of each word in the sequence and apply the softmax activation function [7]. After finishing all the steps our trained model is done so after pass the storyline as an input to the trained model, It helps to generate the sentence based on the first word in the storyline Then after generating first sentence it takes next word and previous sentence as an input and generates next meaningful sentence and so on. As a result the story is generated as an output.

C. Experimental Results

In this framework Storyline is given as an input to trained model. Expected results are written below in tables. Every word in the storyline is considered for sentence formation in story. Then next word and previously formed sentence are used to make prediction of next sentence in story. Storyline helps to make coherent stories.

Table I and II shows the outcomes of generated story for four epochs with respect to the given storyline. Epoch is one of the parameter which we initialize before training a model. One epoch is completed when entire dataset is passed for the forward and backward techniques through neural networks only once.
E.g. in Table I the given storyline as an input is Cook → kids → lunch → noodles → potato salad and the story obtained for four epochs is shown in the TABLE. The story obtained by epoch 1 don't have any meaning and also sentence formulation is not good and there is word repetition in this. Then we check for epoch 5 in this sentence are generated but they are meaningless. For epoch 11 the sentence formulation is very good but there is no connection between two sentences. At last we go for epoch 25 where sentence formulation is very good and also final generated story is coherent.
In the given Table I and II, time and loss is also given. Time is measured in sec and loss computes the error for a single training example.

STORY 1

Story Line	Cook → kids → lunch → noodles → potato salad

EPOCHS	GENERATED STORY	TIME(sec)	TRAIN LOSS
Epoch 1	Ram the cook is is in the form get. Kids loves a a a sentence. The lunch good yet to to done. noodles was not so like. potato salad like it.	687	2.9822
Epoch 5	Ram like cook.The kids play football.Then lunch time is there to. All want noodles also ram it. He makes the potato salad.	526	2.5269
Epoch 11	Ram like to cook. Kids love to play. Sara want lunch. Noodles are tasty. They eat potato salad.	525	2.4701
Epoch 25	Ram liked to cook food items. He take his kids for picnic.Her wife sara serve lunch for them.They like noodles. She serve potato salad with yogurt.	502	1.9498

TABLE I. Story generated by the story line Cook → kids → lunch → noodles → potato salad

ISSN: 2233-7857 IJFGCN

STORY 2

Story Line	Party → cake → balloons → surprise → family

EPOCHS	GENERATED STORY	TIME(sec)	TRAIN LOSS
Epoch 1	Party Plan an special event.The cake not cook cooked.He want co balloons.like surprise much at all.Every family does it.	682	2.9834
Epoch 5	They party plan hard.The cake is just good.every kid balloon.They got surprise.All family together.	530	2.5178
Epoch 11	They plan surprise party. Everyone likes cake.Kids want balloons.They are surprised.family become happy.	518	2.4693
Epoch 25	They Plan surprise Party for daughter.She likes pineapple cake.The room is ull of balloons.She got big surprise.Family became very happy.	498	1.9432

TABLE II. Story generated by the story line Party → cake → balloons → surprise → family

TABLE III shows the How train loss and time vary with respect to epochs. It is observed that as number of epochs goes on increasing train loss and time goes on decreasing.

Epoch	Train loss	Time in sec
1	2.9822	687
2	2.7736	533
3	2.6721	537
4	2.5934	573
5	2.5269	526
6	2.4701	525
7	2.4151	526
8	2.366	538
9	2.321	538
10	2.2781	534
11	2.2387	525
12	2.2009	526
13	2.1626	729
14	2.1298	790
15	2.0957	503
16	2.0615	507

17	2.0327	507
18	2.0043	511
19	1.9748	508
20	1.9498	502
21	1.9263	498
22	1.8991	470
23	1.8699	468
24	1.8523	156
25	1.8365	444
26	1.8122	437
27	1.7948	431
28	1.7689	425
29	1.7377	426
30	1.713	418

TABLE III. Calculation of Train Loss and time by varying epoch hyper parameter of neural network

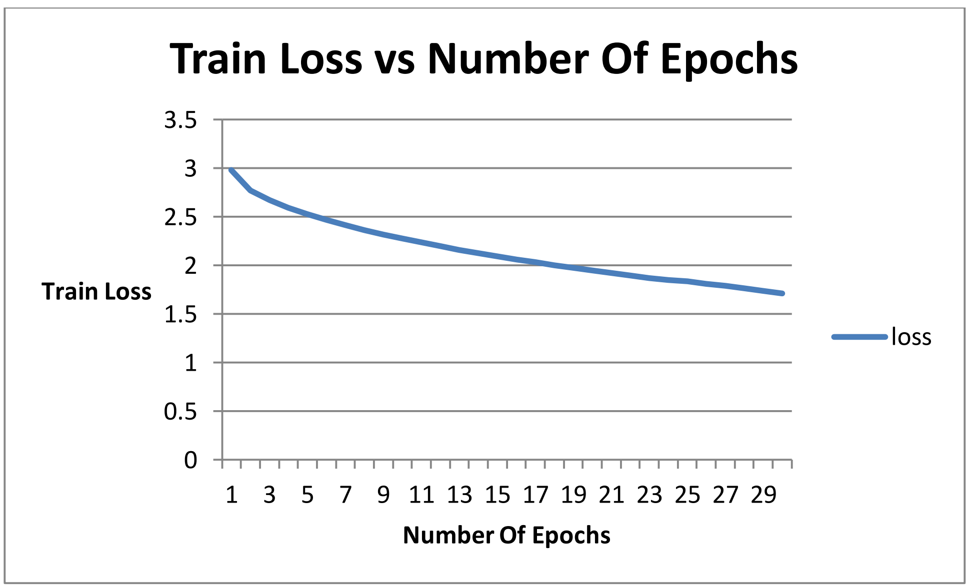

Fig 4.Train loss versus number of epochs

It is observed that as the number of epochs increases the train loss is reduced as shown in Fig.4.The graph shown in above fig.4 is calculated by taking measurement from above TABLE III.

D. Conclusion and Future Work

Up to this stage we conclude that a generated framework first plans a story based on storyline with the help of hierarchical generation. We explore dynamic schema and show that it outperforms the baselines without planning components and tends to generate more coherent and relevant stories with diverse and novel words. The current framework uses a sequence of words to generate a concise and meaningful

story, which simplifies many complex structures in a real story plot. To further improve thematic consistency we plan to extend the study to richer representations, such as entity, event, and relation structures, to describe the story plots.
In future the plan is to extend the framework with multiple RNN hidden layers to see if we can develop a better model or increase accuracy of the model. The framework generates a story which is human-readable text format. It will be more fun and interesting when it will tell stories. Developing such a framework is another goal we would like to achieve.

E. Acknowledgement

The authors would like to thank the Department of Information Technology of Government college of Engineering, Karad, India for their generous support. The authors would like to thank the Principal Dr. A. T. Pise, HoD Dr. S. J. Wagh and all staff members of Information Technology Department for sharing pearls of wisdom with us during the course of this research, and we thank "anonymous" reviewers for their so-called insights. The authors are also immensely grateful to for their comments on an earlier version of the manuscript, although any errors are our own and should not tarnish the reputations of these esteemed persons.

References

1. Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, Rui Yan1, "Plan-and-Write: Towards Better Automatic Storytelling," In The Thirty-Third Association for the Advancement of Artificial Intelligence Conference,2019.
2. Tianming Wang, Xiaojun Wan, "Hierarchical Attention Networks for Sentence Ordering," presented at the Thirty-Third Association for the Advancement of Artificial Intelligence Conference, 2019.
3. Parag Jain, Priyanka, Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, Karthik Sankaranarayanan, "Story Generation from Sequence of Independent Short Descriptions," In proceedings of Workshop on Machine Learning for Creativity, Halifax - Canada, Aug 2017.
4. Angela Fan, Mike Lewis, "Hierarchical Neural Story Generation," Menlo Park, In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 2018.
5. Boyang Li, Stephen Lee-Urban, George Johnston, and Mark O. Riedl, "Story Generation with Crowdsourced Plot Graphs," presented at Association for the Advancement of Artificial Intelligence, 2013.
6. Brent Harrison, Christopher Purdy, Mark O. Riedl, "Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks," presented at the AIIDE-17 Workshop on Intelligent Narrative Technologies WS-17-20, 2017.
7. Melissa Roemmele, "Writing Stories with Help from Recurrent Neural Networks," In proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.
8. Lara J. Martin, PrithvirajAmmanabrolu, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl, "Event Representations for Automated Story Generation with Deep Neural Nets," In Proceedings of SIGKDD Workshop on Machine Learning for Creativity, Halifax, Nova Scotia, Canada, August 2017.
9. Rafael PÉrez Ý Pérez, Mike Sharples, "MEXICA: A computer model of a cognitive account of creative writing," Journal of Experimental & Theoretical Artificial Intelligence, vol. 13,o.2, p. 119-139,[Online document] 11 October 2014. Available: UTSA Libraries, http://dx.doi.org/10.1080/09528130010029820/ .

10. Selmer Bringsjord, David A. Ferrucci, "Artificial Intelligence and Literary Creativity," NEW YORK, CA: Psychology Press, 2013.
11. DiptiPawade, Avani Sakhapara, Mansi Jain, Neha Jain, KrushiGada, "Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM," Modern education and computer science press, I.J. Information Technology and Computer Science, no. 6, p. 44,[Online document] 2018. Available: In MECS, http://www.mecs-press.org/ .
12. Karl Pichotta, Raymond J. Mooney, "Learning Statistical Scripts with LSTM Recurrent Neural Networks," In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.
13. Nanyun Peng, MarjanGhazvininejad, Jonathan May, Kevin Knight, "Towards Controllable Story Generation," In Proceedings of the First Workshop on Storytelling, p. 43–49, Association for Computational Linguistics, June 5, 2018.
14. Melissa Roemmele, Sosuke Kobayashi, Naoya Inoue, Andrew M. Gordon, "An RNN-based Binary Classifier for the Story Cloze Test," In Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, p. 74–80, April 3, 2017.
15. Karl Pichotta, Raymond J. Mooney, "Using Sentence-Level LSTM Language Models for Script Inference," The University of Texas, 8 Jun 2016.
16. Fan A., Lewis M., Dauphin Y., "Hierarchical neural story generation," In ACL, 2018.
17. Montfort N., "Natural language generation and narrative variation in interactive fiction," In AAAI WS, 2006.
18. Juntao Li, Lidong Bing, Lisong Qiu, Dongmin Chen, Dongyan Zhao, Rui Yan, "Learning to Write Stories with Thematic Consistency and Wording Novelty," presented at The Thirty-Third AAAI Conference on Artificial Intelligence, 2019.
19. Cai Z., Tu, L., Gimpel K., "Pay Attention to the Ending: Strong Neural Baselines for the ROC Story Cloze Task," In ACL, p. 616–622, 2017.
20. [20] Martin L., Ammanabrolu P., Hancock W., Singh, S., Harrison B., Riedl, M., "Event Representations for Automated Story Generation with Deep Neural Nets," 2017.
21. Zhang, J., Feng, Y., Wang D., Wang, Y., Abel A., Zhang, S., and Zhang A., "Flexible and Creative Chinese Poetry Generation Using Neural Memory," In ACL, vol. 1, p. 1364–1373. 2017.
22. Swanson, R., and Gordon A. S., "Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling," In ACM Transactions on Interactive Intelligent Systems(TiiS), 2nd ed., vol. 3, no. 16, 2012.

ISSN: 2233-7857 IJFGCN