

Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM

Mrs. Dipti Pawade, Ms. Avani Sakhapara

Assistant Professor, K. J. Somaiya College of Engineering, Department of IT, Mumbai
E-mail: {diptipawade, avanisakhapara}@somaiya.edu

Ms. Mansi Jain, Ms. Neha Jain, Ms. Krushi Gada

Student, K. J. Somaiya College of Engineering, Department of IT, Mumbai
E-mail: {mansi.j, neha.kj, krushi.gada}@somaiya.edu

Received: 27 December 2017; Accepted: 09 March 2018; Published: 08 June 2018

Abstract—With the advent of artificial intelligence, the way technology can assist humans is completely revived. Ranging from finance and medicine to music, gaming, and various other domains, it has slowly become an intricate part of our lives. A neural network, a computer system modeled on the human brain, is one of the methods of implementing artificial intelligence. In this paper, we have implemented a recurrent neural network methodology based text generation system called Story Scrambler. Our system aims to generate a new story based on a series of inputted stories. For new story generation, we have considered two possibilities with respect to nature of inputted stories. Firstly, we have considered the stories with different storyline and characters. Secondly, we have worked with different volumes of the same stories where the storyline is in context with each other and characters are also similar. Results generated by the system are analyzed based on parameters like grammar correctness, linkage of events, interest level and uniqueness.

Index Terms—Recurrent neural networks, Long short-term memory, Text generation, Deep learning.

I. INTRODUCTION

In the field of artificial intelligence, change is real. Deep Learning[6][9] is an emerging field of artificial intelligence, which is now being widely used to identify faces and objects in digital pictures[5], recognize human voice commands in the smartphones and give a response to Internet search queries. This technology can be used to carry out so many other tasks in the future. Currently, it has helped the machines to understand the rules and grammar of the human natural language and the machines are now able to generate text similar to the way, we humans talk and write. Thus Deep Learning can eventually give machines an ability to think, analogous to common sense, which can augment human intelligence [7]. In order to stretch the limits of artificial intelligence and Deep Learning, we have developed Story Scrambler, an automatic text generation tool that will produce a new

story with respect to some previously provided input stories. Story Scrambler finds its applications in various fields, especially where writing is required. For example, Story Scrambler can be useful for creating a sequel to a story or novel or replicating the style of a poet or for making new scenes for a play or TV series and so on.

The primary objective of our Story Scrambler system is to understand the sequence of words and write a new story, similar to human thinking. In order to achieve this objective, there are few challenges. As our system accepts multiple input files and then generates an output story aligned to the subject of input files, understanding the context of each story is very important. Another major issue that needs to be handled is to create semantically as well as a syntactically correct statement. Here maintaining the correlation between different sentences is also a major issue to be handled. We tried to address all these issues by making use of word-level long short-term memory based recurrent neural network.

The rest of the paper is structured as follows. In section II, various approaches for automatic text generation is discussed thoroughly and have summarized it in tabular format (Table 1: Summary of Different Methods for Text Generation). Section III gives insights into the implementation of our approach to develop automatic story scrambler. In section IV we have discussed our observations and finally, in section V, we have concluded stating the worth of our research.

II. RELATED WORK

Chandra et al. [1] in their work have proposed a Natural Language Processing and Deep Learning based algorithmic framework which is demonstrated by generating the context for products on e-commerce websites. Their procedure involves primarily 5 steps carried out in an unsupervised manner which are implemented using extraction and abstraction. The steps include – (a) collecting keywords and description for a particular item for data aggregation (b) avoiding redundancy by removing duplicate and item-specific information by creating a blacklist dictionary (c) filtering

those sentences and paragraphs that satisfy the user's intent (d) using Recurrent Neural Network (RNN) along with Long Short-Term Memory (LSTM) for sentence generation and Skip Grams with Negative Sampling for extraction (e) implementing TextRank so that the output is ordered contextually to provide a coherent read.

In a similar context, Thomaidou et al. [2] have proposed a method to produce promotional text snippets of advertisement content by considering the landing page of the website as their input. In order to articulate a piece of a promotional snippet, they have followed the following steps: (a) information extraction (b) sentiment analysis (c) natural language generation.

Parag Jain et al. [10], discussed a model which takes sequences of short narrations and generates a story out of it. They have used phrased based Statistical Machine Translation to figure out target language phrase-maps and then merge all the phrase-maps to get the target language text. For learning and sequence generation sequence-to-sequence recurrent neural network architecture [14] is implemented.

Boyang Li et al. [11] proposed "SCHEHERAZADE" which is a story generation system using Crowdsourced Plot Graphs. For any activity, a graph is plotted. Each graph consists of three types of nodes viz. normal event, optional event and conditional event connected by precedence constraints and mutual exclusion. Basically, the plot graph is used to generate the story space and based on the precedence constraints and mutual exclusion; the sequence of events is decided. This system claims to provide narrative intelligence but yet there is scope to improve the accuracy of story quality.

Ramesh Nallapati et al. [12] have developed Abstractive Text Summarization model based on encoder-decoder RNN with Attention and Large Vocabulary Trick [13]. They have also used switching decoder/pointer architecture to handle out of vocabulary rare words. Lastly, in order to identify the key sentences, they have tracked the hierarchical document structure with hierarchical attention. For testing purpose, they have used two different databases and have achieved promising results.

Jaya et al. [15] proposed an ontology-based system where a user has to provide characters, objects, and location to generate a story. Using ontology, attributes are provided to the selected parameter and story is generated preserving the context of the domain. Later in 2012 [16], they revised the Proppian's System and put forward a system which is semantically more efficient. To maintain the semantic relevance, each sentence has to go through the reasoning process. This revised Proppian's System is further empowered by the support of ontology for explicit specification of a shared conceptualization.

Huong Thanh Le et al. [17] have put forward text summarization technique which is abstractive in nature. Keyword-based sentence generation is carried using discourse rules [18] and syntactic constraints. After that in order to create summarized and conceptually correct sentence Word graph is used.

Sakhare et al. [19] combined the forte of text

summarization techniques based on the syntactic structure of the sentences and features and recommended a hybrid approach. The input document is first preprocessed. It involves steps like sentence segmentation, tokenization, stop words removal and word stemming. The processed input is then fed separately to syntactic structure based neural network and feature-based neural network. In syntactic structure based neural network approach, first POS tagging is used to get the syntactic structure of the sentence. Then a syntactic tree is constructed using dependency grammar [20]. Once it is done, using backpropagation algorithm the neural network is trained based on two features, viz. frequency of POS tag and sequence of POS tag. In a feature-based neural network, feature score is calculated using multi-layer perceptron feed forward neural network based on features like position data, concept-based feature, numerical data, term weight format based score, title feature, co-relation among sentence and co-relation among paragraph. The final score of the sentence is calculated by taking the weighted average of sentence score obtained using both approaches. Based on the final score ranking is given to each statement and top N ranking statement are considered for the summary preparation.

Selvani et al. [21] have elaborated an extractive method of text summarization based on modified weighting method and modified sentence symmetric method. They have specifically worked on documents in IEEE format which is considered as an input to the system. The following steps are performed in order to get summarized text in the same context:

- Recognize rhetorical roles based on keyword "abstract", "introduction" and "conclusion". Basically, the text extracted from these sections is considered for further processing.
- Split the extracted text into a number of sentences and tokens.
- Perform pre-processing (stemming, stop word removal etc.)
- Calculate sentence score using the proposed algorithm (modified weighting method and modified sentence symmetric method) and consider the highest score sentence as the statement to be included in the summary.

The sentence score calculation using Modified Sentence Symmetric Feature Method is done built on attributes like Cue, Key, Title, Location, Thematic and Emphasize weight of the sentence. On the other hand, the Modified Weighing Method contemplates parameters like word local score, title weight for sentence and sentence to sentence cohesion. They have also compared their results with Top Scored Method [22] and concluded that the proposed algorithm gives better results for precision, recall, and f-feature.

Akif et al. [23] developed a mobile application for summarizing the information on Turkish Wikipedia. They have elaborated the efficiency of hybrid approach when

structural features, Wikipedia based semantic features, and LAS based semantic features are altogether taken into consideration. They considered the length, position, and title as a structural feature. LAS based feature includes a cross method and meaningful word set while Wikipedia based semantic feature consists of Wikipedia keyword count.

Richard et al. [24] described an expert writing prompt

application. Here three random words are selected from WordNet. Concepts of selected words are examined by using ConceptNet and then concept correlation is carried out to set the theme. Characters are set using plot generation. Here iteratively new ConceptNet search is carried out in order to build the full-fledged plot. Lastly, story evaluation is done and if it is not as per the goal, the plot is modified iteratively.

Table 1. Summary of Different Methods for Text Generation

Author	Purpose	Methodology
Chandra et al[1]	To produce context of product from E-commerce site	<ul style="list-style-type: none"> Natural Language Processing Deep Learning-oriented algorithmic framework
Thomaidou et al. [2]	To generate advertising snippet from content of webpage	<ul style="list-style-type: none"> Sentiment analysis Natural language generation
Parag Jain et al. [10]	Generate story from different sequences of narrations	<ul style="list-style-type: none"> Phrased based Statistical Machine Translation Sequence-to-sequence recurrent neural network architecture
Boyang Li et al. [11]	Event-based story generation	<ul style="list-style-type: none"> Crowdsourced Plot Graphs
Ramesh Nallapati et al.[12]	Abstractive Text Summarization	<ul style="list-style-type: none"> Encoder-decoder RNN with Attention Large Vocabulary Trick
Jaya et al. [15]	Story generation according to characters, objects, and location selected by the user	<ul style="list-style-type: none"> Ontology
Jaya et al. [16]	Semantically efficient story generation	<ul style="list-style-type: none"> Revised the Proprian's System Ontology
Huong Thanh Le et al. [17]	Abstractive Text Summarization	<ul style="list-style-type: none"> Discourse rules Syntactic constraints Word graph
Sakhare et al. [19]	Text summarization	<ul style="list-style-type: none"> Ranking based on the sentence score obtained by weighted average of sentence score of syntactic structure based neural network and feature-based neural network
Selvani et al. [21]	Extractive text summarization	<ul style="list-style-type: none"> Modified weighting method Modified sentence symmetric method
Akif et al. [23]	Mobile application for Turkish Wikipedia Text Summarization	<ul style="list-style-type: none"> Hybrid approach based on structural features, LSA and Wikipedia based semantic features
Richard et al. [24]	An expert writing prompt application	<ul style="list-style-type: none"> Knowledge base Inference engine consisting of If..Then rules WordNet ConceptNet Natural Language Processing Concept correlation Plot generation
Anushree Mehta et al. [25]	Generate a story based on the theme and length given by user	<ul style="list-style-type: none"> Natural Language Generation (NLG) process Generate and rank methodology

Anushree Mehta et al. [25] designed a system based on Natural Language Generation (NLG) process and Generate and rank methodology. This system accepts the theme and length of story to be generated as input. The script content is determined by using a lexical database, action graph, Centric Theory (CT) and local coherence. Next step is planning statement using dependency tree. Last but not least using rule-based approach and deep syntactic structure tree story is generated. Table 1 shows the summary of all the discussed methods.

III. IMPLEMENTATION OVERVIEW

Story Scrambler takes different stories as input. Different stories can be having different storylines or same storyline. The system first understands the characters and context of each story and then tries to

build a new storyline. It uses Recurrent Neural Networks - Long Short-Term Memory (RNN-LSTM) [26] to generate a new sequence of text, that is, it forms sentences based on a provided input text. Recurrent Neural Networks (RNN) [4] is dominating complex machine learning problems that involve sequences of inputs. It consists of connections in the form of loops which generates feedback and accordingly learns to change the behavior. Such kind of architecture allows the network to learn and generalize across sequences of inputs instead of identifying individual patterns. We have used RNNs as generative models rather than as predictive models. This means that our network learns the sequence of a problem and then generates an entirely new plausible sequence for the same problem domain.

Long Short-Term Memory (LSTM) [3], a common RNN architecture which can remember values for long or

short durations of time, has been developed to learn sequences of words from multiple input stories which constitute three major modules - input processing, creating and training the neural network and producing a new story.

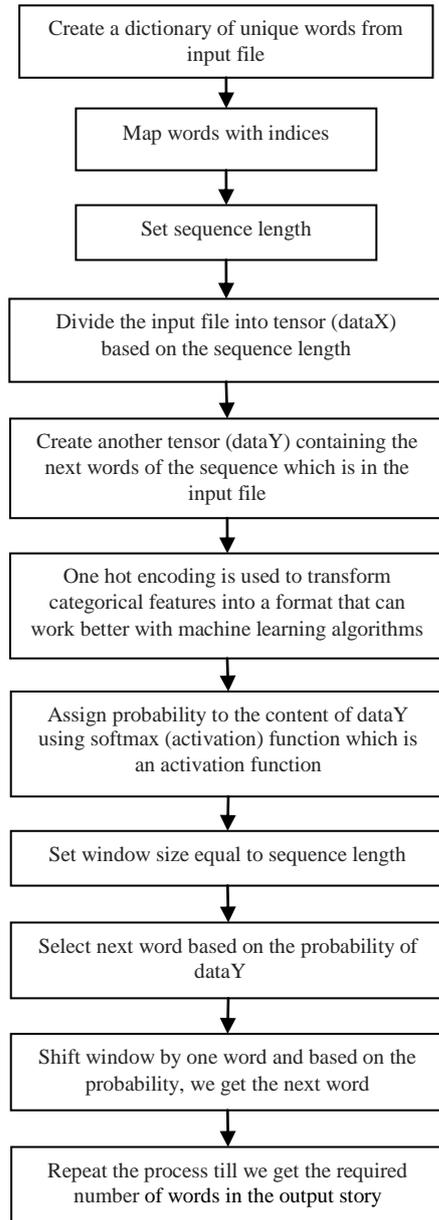


Fig.1. Working of Story Scrambler System

A. Input Processing

This is the first step where we split the input files and the unique words are then stored in a dictionary. Input files can be one or many text files containing stories.

B. Creating and Training the Neural Network

In this module, we create a neural network in which we feed two datasets, dataX which contains words, from the input file, of a particular sequence length; dataY which contains the next word corresponding to the sequence in

dataX. The data in dataX is stored by mapping the individual words of the sequence with their respective indices from the dictionary generated in input processing module. dataX is then reshaped and normalized. The second dataset, dataY, is hot encoded. In this, we create an array where the element at that index which corresponds to the respective word in the dictionary is set as 1 while the elements at the remaining indices are kept as 0.

Now, we define the LSTM model. It will contain 256 memory units (neurons). We then define the hidden layer density.

C. Producing the Story

After training the network, we now produce the new story. A random seed sequence is taken as an input which is considered as the initial window. Based on this sequence, the next word is predicted. The prediction is done by calculating the probability of the words from dataY. This probability is calculated by using the softmax activation function. The window is then updated by adding the newly predicted word to the end.

Once the weights have been assigned, the program now begins generating sentences by selecting words, one at a time. The first word is selected randomly. The subsequent words are decided based on the previously determined word. By iterating over these steps multiple times, we begin forming sentences, paragraphs and ultimately a new story. Fig. 1 gives the detailed overview of the working of our system.

IV. RESULTS AND DISCUSSIONS

Deep Learning is a subfield of machine learning that uses algorithms working similar to brain's neural network, using models called Artificial Neural Networks. These networks are based on a collection of connected units called artificial neurons or neurons. Each connection between the neurons can transmit the signal from one neuron to another. The receiving neuron processes the signal and signals downstream neurons connected to it. The neurons are organized in the form of layers. Different layers may perform a different kind of transformations on inputs. The signals travel through the first layer called as Input Layer, till the last layer called as Output Layer. Any layers in between these two layers are called as Hidden Layer. Some arbitrary weights between 0 and 1 are assigned to each connection between the neurons. For each layer, the weighted sum is computed for the connections that point to the same neuron in the next layer. This sum is then passed to an activation function that transforms the output to a number between 0 and 1. The activation function is similar to the activity of human brains where different neurons are activated by different stimuli. The result of transformation obtained from the activation function is passed to the next neuron in the next layer. This process is repeated till the Output Layer is reached.

During training, the arbitrary weights assigned to each connection between the neurons are constantly updated in

the attempt to reach the optimized values. The optimization of values depends on the optimizer used. One of the widely used optimizers is called Stochastic Gradient Descent (SGD). The objective of SGD is to minimize the loss function. The loss function is the difference between the actual output and the output generated by the model. One single pass of the entire data through the model is called as a single epoch. The model learns through multiple epochs and tries to minimize the loss function. The model is trained by passing the data in batches. A batch is also commonly called as a mini-batch. The batch size refers to the number of data samples that pass through the network at one time.

Deep Learning can be in the form of supervised

learning if labeled data is used and unsupervised learning if unlabeled data is used. For story scrambler system, unsupervised learning is used through Recurrent Neural Networks.

A training set of around 30 stories where each story consists of around 5000 words, is processed on i5 Processor system having 4GB RAM. One tuple in the training set consists of the story divided into a string having the number of words equal to the specified sequence length. The train loss is calculated by varying various parameters like RNN size, the number of layers, batch size and sequence length. These results are shown in Table 2.

Table 2. Calculation of Train Loss by Varying Parameters of the Neural Network

RNN size	No. of layers	Batch size	Sequence length	Train loss
128	1	20	25	2.35
			50	2.1
		100	25	2.34
			50	1.8
	2	20	25	2.45
			50	2.1
		100	25	2.34
			50	1.8
	3	20	25	2.35
			50	2.1
		100	25	2.34
			50	1.8
256	1	20	25	2.2
			50	1.66
		100	25	1.3
			50	0.9
	2	20	25	2.14
			50	1.4
		100	25	2.02
			50	0.741
	3	20	25	2.35
			50	0.71
		100	25	1.96
			50	0.751
512	1	20	25	2.12
			50	0.75
		100	25	2.102
			50	0.34
	2	20	25	1.912
			50	0.71
		100	25	1.8
			50	0.02
	3	20	25	1.67
			50	0.32
		100	25	1.64
			50	0.01

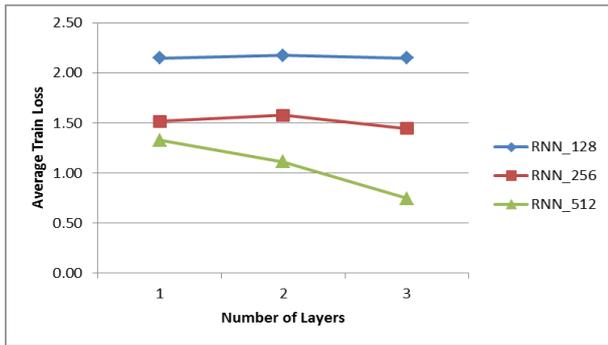


Fig.2. Train loss versus number of layers

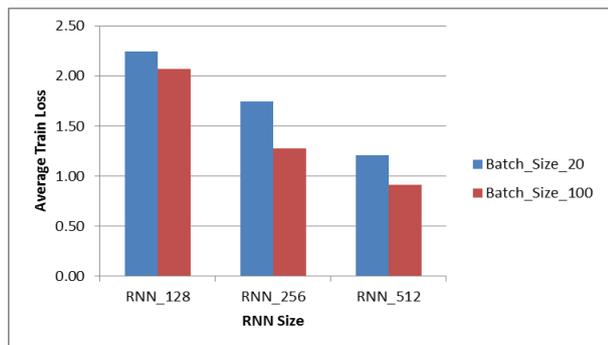


Fig.3. Train loss versus Batch size

It is observed that as the number of layers and the RNN size is increased, the train loss is reduced as shown in Fig. 2. The minimal train loss is obtained for RNN size 512 with 3 layers.

It is also observed that as the batch size increases, the

train loss reduces as shown in Fig. 3. For our system, the minimal train loss is recorded for RNN size 512 with Batch size 100.

Lastly, the sequence length is varied to reduce the train loss further. As shown in Fig. 4, it is observed that as the sequence length is increased, the train loss is decreased.

Finally combining all these results, for our system, the minimal train loss of 0.01 is obtained for RNN size 512 with 3 layers and batch size of 100 having sequence length 50. A good sample story generated from two different input stories is demonstrated in Table 3 and the worst sample story generated is shown in Table 4. Table 5 demonstrates a good sample story generated from two different volumes of the same story. The stories considered are “Alice in Wonderland” [28] and “Charlie and the Chocolate Factory” [27]. Input story 1 and input story 2 in all the demonstrated cases is taken as it is from the book [27][28].

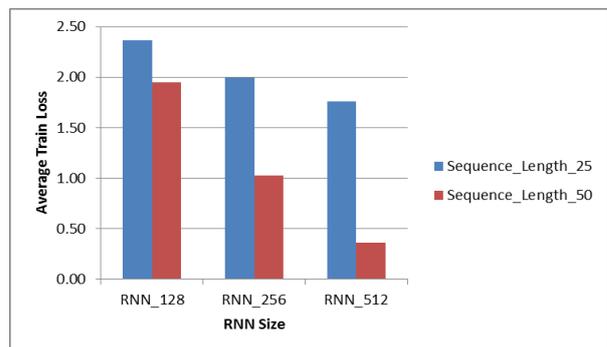


Fig.4. Train loss versus Sequence Length

Table 3. Good Sample Story Generated from Two Different Stories

Input Story 1 [28]	“Alice took up the fan and gloves, and, as the hall was very hot, she kept fanning herself all the time she went on talking: Dear, dear! How queer everything is today! And yesterday things went on just as usual. I wonder if I have been changed in the night? Let me think: was I the same when I got up this morning? I almost think I can remember feeling a little different. But if I am not the same, the next question is, Who in the world am I? Ah, that’s a great puzzle! And she began thinking over all the children she knew that were of the same age as herself, to see if she could have been changed for any of them.” [28]
Input Story 2 [27]	“Walking to school in the mornings, Charlie could see great slabs of chocolate piled up high in the shop windows, and he would stop and stare and press his nose against the glass, his mouth watering like mad. Many times a day, he would see other children taking bars of creamy chocolate out of their pockets and munching them greedily, and that, of course, was pure torture. Only once a year, on his birthday, did Charlie Bucket ever get to taste a bit of chocolate. The whole family saved up their money for that special occasion, and when the great day arrived, Charlie was always presented with one small chocolate bar to eat all by himself.”[27]
Generated Story	This morning Charlie can remember feeling a little different. Today he would see other children of the same age as herself, taking bars of creamy chocolate piled up high in the shop windows. He would stop and think. Ah, that’s a great puzzle! His mouth watering like mad.

Table 4. Worst Sample Story Generated from Two Different Stories

Input Story 1 [28]	“Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, what is the use of a book, thought Alice a book without pictures or conversations? So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her. There was nothing so very remarkable in that; nor did Alice think it so very much out of the way to hear the Rabbit say to itself, Oh dear! Oh dear! I shall be late! Alice (when she thought it over afterward, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural)”[28]
Input Story 2 [27]	“Here Comes Charlie. These two very old people are the father and mother of Mr. Bucket. Their names are Grandpa Joe and Grandma Josephine. And these two very old people are the father and mother of Mrs. Bucket. Their names are Grandpa George and Grandma Georgina. This is Mr. Bucket. This is Mrs. Bucket. Mr. and Mrs. Bucket have a small boy whose name is Charlie. This is Charlie. How do you do? And how did you do? And how do you do again? He is pleased to meet you. The whole of this family had the six grownups (count them) and little Charlie Bucket all living together in a small wooden house on the edge of a great town. The house wasn't nearly large enough for so many people, and life was extremely uncomfortable for them all. There were only two rooms in the place altogether, and there was only one bed.”[27]
Generated Story	Mr. and Mrs. Bucket have wondered. These two very old people are the father and of having nothing to do. You do again? When suddenly a small boy whose name is the use of a book her sister was reading. The whole of this family had no pictures or conversations in it.

Table 5. Good Sample Story Generated from Different Volumes of the Same Story

Input Story 1 [27]	“Oh, what a man he is, this Mr. Willy Wonka! cried, Grandpa Joe. 'Did you know, for example, that he has himself invented more than two hundred new kinds of chocolate bars, each with a different center, each far sweeter and creamier and more delicious than anything the other chocolate factories can make! Perfectly true!' cried Grandma Josephine. 'And he sends them to all the four corners of the earth! Isn't that so, Grandpa Joe? It is, my dear, it is. And to all the kings and presidents of the world as well. But it isn't only chocolate bars that he makes. Oh, dear me, no! He has some really fantastic inventions up his sleeve, Mr. Willy Wonka has! Did you know that he has invented a way of making chocolate ice cream so that it stays cold for hours and hours without being in the refrigerator? You can even leave it lying in the sun all morning on a hot day and it won't go runny!’” [27]
Input Story 2 [27]	“The old man was sitting in the back of the boat and little Charlie Bucket was right beside him. Charlie was holding tightly on to his grandfather's bony old hand. He was in a whirl of excitement. Everything that he had seen so far, the great chocolate river, the waterfall, the huge sucking pipes, the minty sugar meadows, the Oompa-Loompas, the beautiful pink boat, and most of all, Mr Willy Wonka himself had been so astonishing that he began to wonder whether there could possibly be any more astonishments left. Where were they going now? What were they going to see? And what in the world was going to happen in the next room? Isn't it marvelous?” said Grandpa Joe, grinning at Charlie. Charlie nodded and smiled up at the old man. Suddenly, Mr. Wonka, who was sitting on Charlie's other side, reached down into the bottom of the boat, picked up a large mug, dipped it into the river, filled it with chocolate, and handed it to Charlie.” [27]
Generated Story	Mr. Willy Wonka was in a whirl of excitement. He had seen so far the four corners of the earth! He has himself invented a way of making chocolate river. He sends them to Oompa-Loompas, the kings and presidents of the world. Isn't it marvelous? cried Grandma Josephine

Through experimentation, it is also observed that increasing the number of layers beyond three and batch size beyond 100 results into overfitting. Overfitting occurs when the model works well with all the data in the training set but fails to work with the data which is not there in training set.

To measure the accuracy of our system, the generated stories are evaluated by 7 people (H1-H7) for examining the grammar correctness, linkage of events, level of interest and uniqueness in the generated stories. The human rating values are 0 (zero), 25, 50, 75 and 95,

where each of these values indicates the percentage of the aspect to be examined in the generated story. A sample human rating for examining the linkage of events for 5 generated stories is shown in Table 6. Out of 5 generated stories, the stories 1 and 2 are generated by inputting different stories whereas stories 3, 4 and 5 are generated by inputting different volumes of the same story. Only the stories having average human evaluation percentage greater than 60 for all the aspects are accepted as correctly generated stories.

Table 6. Human Evaluation of Generated Stories for examining the linkage of events

		Generated Stories				
		Stories Generated from Different Stories		Stories Generated from Different Volumes of the Same Story		
		1	2	3	4	5
Human Rating	H1	75	0	50	0	95
	H2	95	25	75	50	50
	H3	75	0	95	25	95
	H4	95	25	95	50	25
	H5	75	0	95	50	75
	H6	75	25	50	0	50
	H7	50	0	50	25	50
Average		77.14	10.71	72.86	28.57	62.86
Interpretation		Accept	Reject	Accept	Reject	Accept

In Table 7, the average of the human evaluation for all the aspects is given for the 5 sample generated stories and its observations and interpretation is discussed. From table 7, the story numbers 1, 3 and 5 are accepted as good generated stories as they all have average human rating percentage greater than 60 for all the four aspects considered. It is also observed that the linkage of events and grammar correctness plays a major role in the generation of a good story. In our system, we are able to

achieve good linkage of events for stories generated from different stories as well, since the story generation in our system depends on the number of common words in the inputted stories. The interest level is directly proportional to the linkage of events. Uniqueness is inherently achieved for the stories generated from different input stories whereas the stories generated from different volumes of the same story strive for uniqueness.

Table 7. Average Human Evaluation for examining the grammar correctness, linkage of events, interest, and uniqueness

	Generated Stories					Stories Accepted
	Stories Generated from Different Stories		Stories Generated from Different Volumes of the Same Story			
	1	2	3	4	5	
Grammar	67.32	34.86	71.14	63.51	68.63	1,3,4,5
Events linkage	77.14	10.71	72.86	28.57	62.85	1,3,5
Interest Level	73.59	12.38	67.42	14.73	64.49	1,3,5
Uniqueness	78.15	28.92	63.26	21.34	62.12	1,3,5

V. CONCLUSION AND FUTURE WORK

In this paper, an implementation of story scrambler system using RNN and LSTM is discussed. By increasing the values of different parameters such as the number of neurons, number of layers, batch size and sequence length, we have tried to minimize the train loss. The stories formed are also evaluated by humans and an accuracy of 63% is obtained. The accuracy of the system can be improved further by considering the contextual meaning of the words. Also, synonyms can be used to further improve the accuracy of the system. This system can be further extended for the automatic generation of messages or news articles or jokes or posts.

REFERENCES

[1] Chandra Khatri, Sumanvoleti, Sathish Veeraghavan, Nish Parikh, Atiq Islam, Shifa Mahmood, Neeraj Garg, and Vivek Singh, "Algorithmic Content Generation for Products", Proceedings of IEEE International Conference on Big Data, Santa Clara, CA(2015), pp.2945-2947.
 [2] S. Thomaidou, I. Lourentzou, P. Katsivelis-Perakis, and M. Vazirgiannis, "Automated Snippet Generation for Online Advertising", Proceedings of ACM International

Conference on Information and Knowledge Management (CIKM'13), San Francisco, USA, (2013), pp.1841- 1844.
 [3] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large-Scale Acoustic Modeling". In Fifteenth Annual Conference of the International Speech Communication Association (2014).
 [4] Andrej Karpathy, Justin Johnson, and Li Fei-Fei, "Visualizing and Understanding Recurrent Networks". arXiv preprint arXiv:1506.02078 (2015).
 [5] <https://www.wired.com/2013/02/android-neural-network/>
 [6] Neural Networks and Deep Learning. Retrieved from: <http://neuralnetworksanddeeplearning.com/>
 [7] Ilya Sutskever, James Martens, and Geoffrey Hinton, "Generating Text with Recurrent Neural Networks", Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, (2011), pp.1017-1024
 [8] Martens, J. "Deep learning via Hessian-free optimization", Proceedings of 27 International Conference on Machine Learning (ICML) (2010), pp.735-742.
 [9] Soniya, Sandeep Paul, Lotika Singh, "A review on advances in deep learning", Proceeding of international conference on Computational Intelligence: Theories, Applications and Future Directions (WCI), Kanpur, India,(14-17 Dec. 2015).
 [10] 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 (SIGKDD'17)(Aug 2017), Halifax – Canada.
- [11] Boyang Li, Stephen Lee-Urban, George Johnston, and Mark O. Riedl, "Story Generation with Crowdsourced Plot Graphs" In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, Washington (14 – 18 July 2013) Pp 598-604.
- [12] Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, Bing Xiang, "Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond", The SIGNLL Conference on Computational Natural Language Learning (CoNLL)(2016).
- [13] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, "End-to-end attention-based large vocabulary speech recognition" CoRR, abs/1508.04395 (2015).
- [14] Ilya Sutskever, Oriol Vinyals, and Qoc V. Le., "Sequence to Sequence Learning with Neural Networks", CoRR (2014).
- [15] Jaya A.Uma G.V., "An intelligent system for semi-automatic story generation for kids using ontology" Proceedings of the 3rd Annual Computer Conference, Compute (22-23 Jan 2010), Bangalore, India.
- [16] Jaya A.Uma G.V., "An Intelligent Automatic Story Generation System by Revising Propian's System." Communications in Computer and Information Science (2011) vol 131. Springer, Berlin, Heidelberg
- [17] Huong Thanh Le; Tien Manh Le, "An approach to Abstractive Text Summarization", In proceeding of International Conference of Soft Computing and Pattern Recognition (SoCPaR), (15-18 Dec 2013), Hanoi, Vietnam.
- [18] Le, H.T., Abeyasinghe, G. and Huyck, C., "Generating Discourse Structures for Written Texts," In Proc. of COLING (2004), Switzerland.
- [19] D.Y. Sakhare, Dr. Raj Kumar, "Syntactic and Sentence Feature Based Hybrid Approach for Text Summarization", I.J. Information Technology and Computer Science, 2014, 03, 38-46, DOI: 10.5815/ijitcs.2014.03.05
- [20] Joakim Nivre, "Dependency Grammar and Dependency Parsing", In MSI report 05133, 2005.
- [21] Selvani Deepthi Kavila, Radhika Y, "Extractive Text Summarization Using Modified Weighing and Sentence Symmetric Feature Methods", IJMECS, vol.7, no.10, pp.33-39, 2015.DOI: 10.5815/ijmeecs.2015.10.05
- [22] Maryam Kiabod, Mohammad Naderi Dehkordi and Sayed Mehran Sharafi, "A Novel Method of Significant Words Identification in Text Summarization", Journal Of Emerging Technologies In Web Intelligence, VOL. 4, NO. 3, August 2012
- [23] Akif Hatipoglu, Sevinç İlhan Omurca, "A Turkish Wikipedia Text Summarization System for Mobile Devices," I.J. Information Technology and Computer Science, 2016, 01, 1-10 DOI: 10.5815/ijitcs.2016.01.01
- [24] Richard S. Colon; Prabir K. Patra; Khaled M. Elleithy, "Random Word Retrieval for Automatic StoryGeneration" Proceedings of Zone I Conference of the American Society for Engineering Education (ASEE Zone 1) (3-5 April 2014), Copenhagen, Denmark.
- [25] Anushree Mehta; Resham Gala; Lakshmi Kurup, "A roadmap to auto story generation", 2016 International Conference on Computing for Sustainable Global Development (INDIACom), (16-18 March 2016), New Delhi, India.
- [26] Dipti Pawade, Mansi Jain, Gauri Sarode, "Methods For Automatic Text Generation", i-manager's Journal on Computer Science, Volume 4. No. 4, December 2016 - February 2017, Pp. 32-36.
- [27] Roald Dahl, "Charlie and the Chocolate Factory".
- [28] Lewis Carroll, "Alice in Wonderland"

Authors' Profiles



Dipti Yogesh Pawade received B.E. degree in Computer Science and Engineering from Sant Gadge Baba University in 2009 and M.E. degree in Embedded System and Computing from G. H. Raisoni College of Engineering, Nagpur in 2012. Since 2012 she is an Assistant Professor in the Department of Information Technology at K. J. Somaiya College of Engineering, Vidyavihar, Mumbai. Her interest includes Machine learning, Web Security, and Web Application Development.



Avani Sakhapara received B.E. degree in Information Technology from Thakur College of Engineering and Technology, Mumbai University and M.E degree in Computer Engineering from K.J. Somaiya College of Engineering, Mumbai University. Since 2009 she is an Assistant Professor in the Department of Information Technology at K.J. Somaiya College of Engineering, Vidyavihar, Mumbai. Her interest includes Information Security, Machine Learning, Natural Language Processing and Web Application Development.



Mansi Jain received the B.E. degree in Information Technology from K. J. Somaiya College of Engineering, Vidyavihar, an Autonomous college affiliated to Mumbai University in 2017. Her interest includes Artificial Intelligence, Computer Vision and Algorithms.



Neha Jain received the B.E. degree in Information Technology from K. J. Somaiya College of Engineering, Vidyavihar, an Autonomous college affiliated to Mumbai University in 2017. Her interest includes Deep learning, application development, and data structures.



Krushi Gada received the B.E. degree in Information Technology from K. J. Somaiya College of Engineering, Vidyavihar, an Autonomous college affiliated to Mumbai University in 2017. Her interest includes Competitive Programming, Discrete Structures, and Artificial Intelligence.

How to cite this paper: Dipti Pawade, Avani Sakhapara, Mansi Jain, Neha Jain, Krushi Gada, "Story Scrambler - Automatic Text Generation Using Word Level RNN-LSTM", International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.6, pp.44-53, 2018. DOI: 10.5815/ijitcs.2018.06.05