The Pennsylvania State University
The Graduate School

STORY GENERATION USING HIERARCHICAL CONVOLUTIONAL NETWORKS

A Thesis in
Computer Science and Engineering
by
Saniya Naphade

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2021

The thesis of Saniya Naphade was reviewed and approved by the following:

Clyde Lee Giles
Professor of Computer Science and Engineering
Thesis Co-Advisor

Ting-Hao (Kenneth) Huang
Assistant Professor of College of Information Sciences and Technology
Thesis Co-Advisor

Rui Zhang
Assistant Professor of Computer Science and Engineering

Chita R. Das
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

Abstract

Many writers suffer from writer's block when composing long stories with no fixed direction. A creative artificial intelligence system that can generate well reasoned and eloquent texts with limited inputs can provide valuable insights. This type of system can help writers to explore various directions in which the story could progress. However, the automated story generation task is usually experimented with toy datasets comprising of disconnected topics. These toy datasets usually contain artificial instances and topics (usually short sentences) making it easier for the model to learn and generate coherent passages. This work explores whether hierarchical story generation model can construct fluent follow-up passages depending on the instance from a real human story as input. All the instances are generated using the real-stories from the BookCorpus dataset. The instances are constructed by using extractive summarization on portions of the real story before being given as input Human readers have ranked the generated passages higher than the ground truth passages.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my co-advisor Dr. Ting-Hao (Kenneth) Huang from the College of Information Science and Technology. Without Dr. Huang's support and guidance it would have been really difficult for me to complete my thesis within the specified timeline. His consistent encouragement and feedback helped me to remain on track and continue to make progress in the correct direction. I would also like to extend my gratitude towards my co-advisor, Dr. C. Lee Giles, of Department of Computer Science and Engineering, for his feedback and suggestions. I would also like to express my token of gratitude towards Dr. Rui Zhang from Computer Science and Engineering.
A special token to thanks to Chieh-Yang Huang and Kavya Laalasa Karanam from the Crowd-AI Lab at Pennsylvania State University for their constant support and motivation.

Chapter 1
Introduction

Storytelling is more than just an art form; it is a fundamental component of what makes us human. We can share information through stories while also forming an emotional connection. Because of its significance, much research has been conducted to discover techniques to combat writer's block and supplement human writing. Most of these techniques center around fundamental language support, such as sentence completion, detecting grammatical errors, and spell-checking. Although the majority of these techniques are simple, they have proven to be extremely useful in the field of literature. However, more complex language support such as sentence generation while maintaining thematic consistency over the entire document while establishing very long range dependencies is still in its early experimental stages.

This task is experimented with toy datasets which contain primitive instances that can help the model learn. Datasets such as ROCStories [1] contain extremely short stories usually containing five to seven sentences or GLUCOSE [2] which comprises of stories with various constraints allowing easier inferences making it simpler for a language model to train. Datasets such as WritingPrompts [3] have also been developed with cater more towards the task of story generation. However, this dataset consists of well-composed and unique prompts or premises which is usually not the case for human writers.

The work in this thesis focuses on exploring story generation models' ability to produce fluent and thematically consistent follow-up stories for a given portion of human-written story. The human-written story is divided into story blocks of fixed sentence lengths(in this case, 20 sentences each). The model is then trained learn the current story block and generate a story for the next story block.

This work focuses on the extension of the methodology proposed in Hierarchical Neural Story Generation [3]. This work [3] propose a hierarchical system for generation of coherent and fluent stories. In this method, first a prompts or premise outlining the

general theme of the story is constructed and then a sequence2sequence model is trained on this premise to generate a story. This type of modelling on a prompt makes it easier for the sequence model to generate consistent stories as the prompt or premise establish an overall plotline of the story. It also curbs the tendency of the sequence model to drift off-topic.

We experiment with this framework by first creating a prompt or premise from the current story block and then using a convolutional sequence2sequence model to condition on that prompt to generate a passage for the next succeeding story block.

The model is trained and evaluated on story blocks created human-written stories encompassed in the BookCorpus dataset [4]. The results of the experiment show that the stories generated using this method [3] outperform the models from Plan and Write [5] and GPT-2 [5], while suffering in a semantic-based metric. Human evaluation results have shown that readers prefer the long stories generated by the hierarchical method over stories generated by the models proposed in Plan and Write [5] and GPT-2 [5].

Chapter 2
Related Work

2.1 Story Generation

For various text generation tasks such as machine translation, summarization tasks, state of the art results have been obtained using sequence-to-sequence models. The early works in story generation focus on establishing coherent event sequences [6]. There has also been significant work done in generating short stories using case-based reasoning [7, 8]. For writing Wikipedia articles [9], the authors introduce a decoder-only architecture to generate long sequence of sentences. While for Chinese poetry generation [10] makes use of recurrent neural networks.

Most previous works in story generation focus on using recurrent neural networks dependent sequence-to-sequence [11] modelling using various sources as inspiration for story generation. Like [12] makes use of images as inspiration for generation of short paragraphs of text. This model is trained on romance novels in particular. While work [13] relies on building effective connections between independent events to weave a story. There has also been work in two step generation such as showcased in [14], where in first a set of events are established from the text and then this sequence of events is used for modelling stories. Another work similar to this is by Harrison et al.(2017) [15], wherein RNNs are used to generate summaries of movies which form a sequence of events. Then, Markov Chain Monte Carlo sampling of the event sequences are used to generate stories. The work described in [16] makes use of ensemble-based technique to generate semantically correct sequences of text for the story. The three step architecture described in [17] further improves on the task of generating semantically coherent sentences. In this work, predicate-argument structure to improve the consistency of the generated story. The task of story generation has been viewed as a lab experiment with the models being trained on dummy or toy datasets specifically created for the task,

such as WritingPrompts [3], ROCStories [1], or WikiPlots [18]. There hasn't been much work where real life stories are used for the generation task.

Chapter 3
Dataset Preprocessing and Prompt Generation

3.1 Introduction

For this project, we focus on the BookCorpus dataset [4]. The dataset consists of real human-written stories scrapped from Google Books website. The dataset consists of around $15,605$ raw fiction books. Before forming the story blocks of 20 sentences each, the stories were cleaned to extract relevant information. The rules for cleaning of the text are based in the work by Huang and Huang [19].

3.2 Dataset Preprocessing

The dataset is cleaned to extract relevant information i.e, the story content. The following rules are used for cleaning the dataset:

- remove non-fiction material such as publication information, acknowledgement
- short story books of size less than 10 KB
- non-english stories
- stories in e-book formats
- stories belonging to genres other than fiction

All the non-fiction portion of the stories was removed using regular expressions. This includes all information before the first chapter and after the last chapter. However,

since it is difficult to ascertain the end of a book, we eliminate the final chapter as well. This is accomplished by using regular expressions to find the term "Chapter". After thorough cleaning, a total of $4,794$ books are in the dataset which are then split into train, validation and test sets. The split is in the accordance to the ratio of $70:10:20$, which means, a total of $3,357$ books in the train set, 479 books in the validation set and 958 book sin the test set.

In order to set for continuation in the generation of text, the stories were divided into blocks of around 20 sentences each. This helped in increasing the size of the train, validation, and test sets, with more than $90,000$ instances in the train set and around $250,000$ instances in the test set. For human evaluation, one test instance is selected from each book from the test set, leaving around 200 selected instances. And around 1000 instances are used for automatic evaluation

# Training Books	$3,357$
# Validation Books	479
# Testing Books	958
# Training prompts	$921,421$
# Validation prompts	$1,259$
# Testing prompts	$282,469$

Table 3.1. Dataset statistics

3.3 Premise Generation

The quality of the story generated highly depends on the quality of the prompt or premise. With a premise being short of containing only about 1000 words, a 20-sentence story block is a huge input, however, taking only a few sentences from the story block fails to provide the model with enough context to generate concise stories. To deal with this problem, ten previous story blocks are concatenated and a summary is generated of the concatenated story blocks to generate the prompt or premise as seen in figure 3.1. The number ten is chosen after trial-and-error where the other selected numbers were three, five, and fifteen. The formulation of this problem would be, a short summary of previous ten story blocks is taken as input to produce a 150-200 tokens long story. All the gold stories are truncated to have around 1000 words.The vocabulary size is limited to words that appear more than 10 times each in both prompts and stories. An end of document token and unknown word token are also included.

The summary of the 200 sentence stories is generated using PacSum [20] method is used. This extractive summarization model uses unsupervised learning algorithm to generate summaries.The summarization model is adapted to this particular story generation task, the score weighting algorithm of the model is fine-tuned using the Shmoop corpus [21]. This dataset consists of around 200 stories with multi-paragraph summaries for each individual chapter.

As we are concatenating 10 story blocks, the first premise that would be generated would be for story block, $B11$.

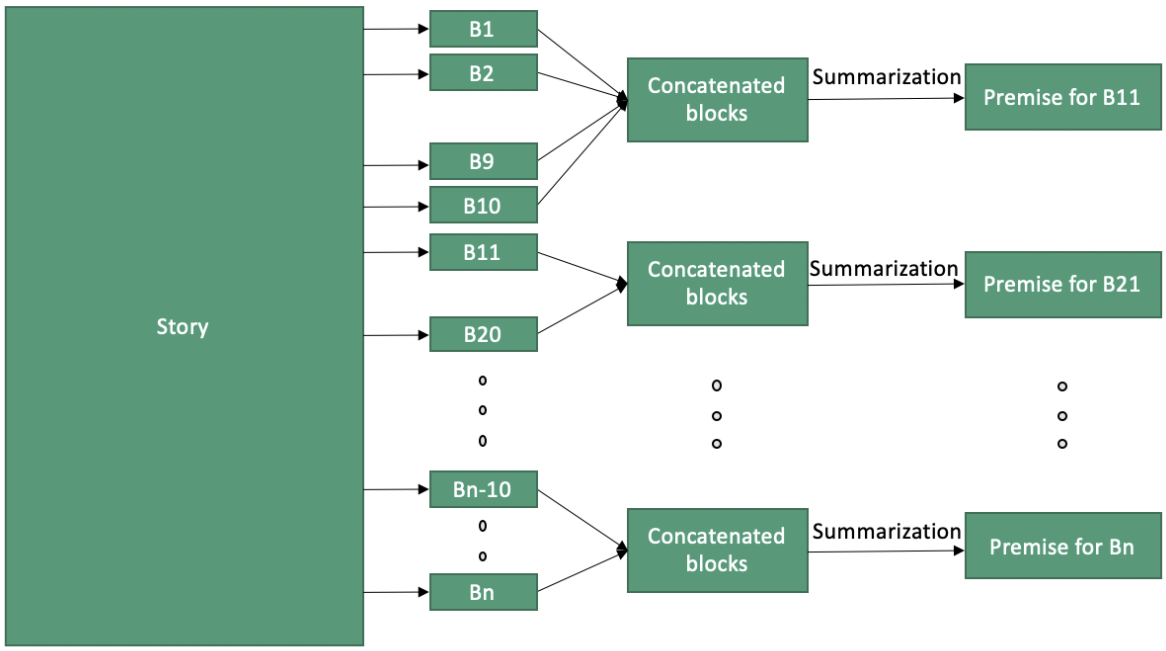

Figure 3.1. Premise Generation for future blocks. Previous 10 story blocks denoted as $B1, B2, ..$ are concatenated and summarized to form a premise for the future Block $Bn + 10$

Chapter 4
Methodology

The work described in the paper, Hierarchical Neural Story Generation [3], is expanded to real-world, human-written stories. The work describes method to establish long-term dependencies and generation of multi-sentence stories. Although previous works in long sentence generation using recurrent [22] neural networks and convolutional networks [23] have shown significant success, the problem of generation of long sequences of text is still a problem. The work described in the paper [3], provides insight into this problem of generation of long sequences of text without losing sight of the initial premise. This thesis is the expansion of the work described in the paper [3].

4.1 Convolutional Seq2Seq model

Given the length of the stories in the dataset, BookCorpus [4], using RNNs would be a challenge for processing the tokens sequentially. Moreover, RNNs over would suffer from decaying gradients due to large length of the sentences. In place of RNNs, convolutional sequence-to-sequence models introduced in the paper, Convolutional sequence to sequence models [23]. In this model, the traditional RNNs or LSTMs in the encoders and decoders of the sequence-to-sequence models are replaced with convolutional deep networks. Convolutional sequence-to-sequence models are more suitable for generating long sentences as the sentences to be processed in parallel, making the process faster and more efficient. The encoders and decoders of the sequence-to-sequence model are connected to the attention module which helps in providing attention to the required terms in the decoder layer using the encoder outputs.

4.1.1 Attention Mechanism

Convolutional Neural Networks are able to model limited context depending on the window size. This prevents the model from learning long-term dependencies when producing the output story. This is counteracted by introducing self-attention mechanism [24] in the decoder This attention mechanism allows the decoder to track and refer to any of its previously generated words. This helps the model to reduce redundancy and repetition. This is accomplished by using Gated Linear Units.

Gated Linear Units: Unlike the attention mechanism which takes the linear projections of the queries, keys and values to calculate the attention, explored in the paper, Attention is All You Need [24], the self-attention is calculated using deep neural networks with *gated linear units* [25].

The gated linear unit [25] mechanism uses a sigmoid function which acts as gate. This gate is an integral part when finding the most relevant portion of the input for the given context. This allows the model to find the most relevant words and in a sense fine-tune, when predicting the next word.

Multi-Scale Attention: Along with multi-head attention [24], in this work multi-scale attention is also used. This allows different attention heads to see different information unlike the multi-head attention mechanism in which all the attention heads see the same input. As each attention head sees different input, the heads are not redundant as each head learns something different. This is accomplished by downsampling the input to each attention ahead by different amounts. Also unlike the self-attention mechanism in [24], only the past timesteps are attended to and not the current one.

4.2 Fusion Model for condition on Premise

The story generation task is a very open-ended problem unlike any translation task where the semantics of the target are completely specified by the source. In this case, the sequence-to-sequence models tend to ignore the premise and focus more on the generation of the text. The fusion mechanism proposed is similar to the cold fusion mechanism proposed in Cold fusion: Training seq2seq models together with language models [26]. In this fusion mechanism, a sequence-to-sequence model is first trained on the premise, following which another sequence-to-sequence model is then trained with access to the hidden states of the previously trained sequence-to-sequence model. This helps in conditioning on the premise as the model can always refer back to the original

premise to stay on track. This sort of conditioning allows the model to generate stories without digressing from the original premise laid out.

This is achieved by first concatenating the hidden states of both the sequence-to-sequence models and passing the concatenated state through two individual gates. The two separate gates help to decide what information from the two models independently should be used. These gated hidden states are then again concatenated and the combined state is then passed through multiple fully connected gated linear units and softmaxed to predict the next word. Due to the access to the hidden state of the pretrained model, the training model can focus on learning what the pretrained model might have missed, i.e, condition on the prompt.

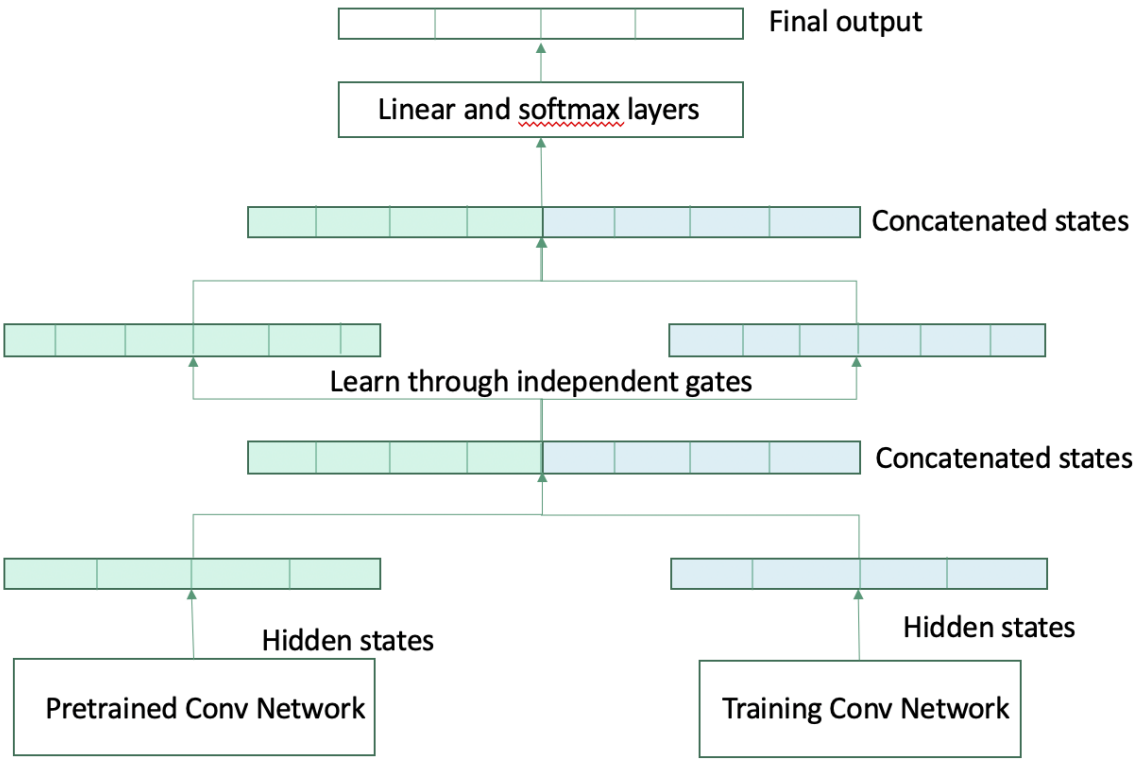

Figure 4.1. Architecture of Fusion Model

Chapter 5
Experimentation

To quantify the performance of the generated story various automatic evaluation metrics have been used. However, the automatic metrics don't reflect the quality of the generated stories accurately. To effectively judge the quality of the generated story, human evaluations are also performed.
The following baseline models have been evaluated on for evaluations:

- Ground Truth:The target story for $Block_{n+1}$
- Random_Future: The story block selected randomly between $Block_n + 5$ to $Block_{n+15}$
- Random_History: A randomly selected block prior to $Block_{n-10}$
- Plan&Write: The story generated for $Block_{n+1}$ using an intermediate plot representation with $Block_n$ as an input to the model [5]
- GPT2: The story generated for the next story block with the current block as input using Sematic Frame Represenations [19]

The models Plan&Write and GPT2 are the works from collaborators working on the same problem but with different approaches. As human evaluations is an expensive task and this being an explorative project, the testing instances for the evaluation set are worked in collaboration with my lab mates, ChiehYang Huang(contributor of the GPT2 model) and Kavya Laalasa Karanam(contributor of the Plan&Write model). Also, we in collaboration have submitted a short paper for publication in the EMNLP2021 conference. Due to this, the results in the tables 5.1, 5.2 and 5.3 are identical to those is the work submitted by Kavya [27].

5.1 Automatic Evaluation

For automatic evaluations we use both word-overlapping and context based evaluation metrics.The commonly used word-overlapping metrics for machine translation tasks such as BLEU-4, METEOR and ROUGE-L have been used for evaluating the model performance. While for context-based metrics, SkipThought Cosine Similarity score has been used. Out of the $280,000$ instance, a total of 956 instances have randomly been selected for automatic evaluations. All the 958 generated stories are compared with the gold targets for both word-overlapping and context-based evaluations. From Table 5.1, it is observed that the stories generated by the sequence-to-sequence model outperform all the other models for word-overlapping metrics. While its performance for the context-based evaluation is lagging behind that of GPT2 and Plan&Write, it is still reasonable and tolerable. The work by Fan et al.(2018) [3], states that automatic evalaution metrics are not suitable for this kind of an open-ended research problem, as we are more concerned in knowing whether the model is capable for generating practical and interesting stories and not a specific story. This is also further explored in the works by Hsu et at.(2019) [28].

Metrics	Random_future	Random_History	Conv_Seq2Seq	Plan&Write	GPT2
BLEU-4	0.0002	0.0001	0.0026	0.0000	0.0002
ROUGE-L	0.0763	0.0739	0.1414	0.0732	0.0864
METEOR	0.0306	0.0316	0.1014	0.0315	0.0352
CosineSimilarity	0.5525	0.5595	0.3436	0.5586	0.5348

Table 5.1. Automatic Evaluation results based on both word-overlapping metrics, namely BLEU-4, ROUGE-L and METEOR and context-based, CosineSimilarity scores for the generated stories

5.2 Human Evaluation

To properly validate the generation results of this model, hum evaluations are conducted. Amazon Mechanical Turk(MTurk) has been used to accomplish this task. Two experiments are desgined to check the models' generation abilities.

- 3-sentence stories
- 200-word stories

5.2.1 3-Sentence Stories

All the previously mentioned 6 models except the Convolution sequence-to-sequence model being discussed in this work. This is because the convolution sequence-to-sequence model produces long stories of around 150-200 words.

Random_Future is considered to be the strong baseline and Random_History to be the weak baseline for this experiment. The results obtained are follow:

Model	Average Ranking	Standard Deviation
Ground_Truth	3.432	1.646
Random_Future	3.620	1.552
Random_History	3.564	1.544
Plan&Write	4.372	1.624
GPT2	3.446	1.543
Conv_Seq2Seq	2.566	1.831

Table 5.2. Human Evaluation Results for 3-sentence stories generated by the model

The 3-sentence stories for Ground_truth, Random_Future and Random_History are PacSum [20] generated 3-sentence long stories over the target story block for Ground_Truth, a randomly selected block between $Block_n + 5$ to $Block_{n+15}$ for Random_Future, and a randomly selected block prior to $Block_{n10}$ for Random_History.

For this Human Intelligence(HIT) task, all the workers are given the story block in question, $Block_n$ and the stories generated by all the stories generated by the 6 models. MTurk workers are asked to rank the story ideas from one to six with one being the highest possible rank and six being the lowest. The workers are urged to judge the six generated stories on the plot ideas rather than focusing on grammatical mistakes or minor repetitions.

A fixed time of 30 seconds is allotted for each instance allowing the workers with enough time to read the stories. Every task is completed with five different workers with each task taking less than two minutes to finish, estimating a total of $0.33 per HIT. A total of 100 instances are used for this task.

All recruited workers meet the following qualification, (*i*) Number of Approved HITs(3000), (*ii*) HIT Approval Rate(98%), (*iii*) US-Locale, and (*iv*) Adult Content Qualification.

It is observed based on the results reported in 5.2, the human readers favor the longer stories generated by convolutional sequence-to-sequence model over even the Ground-Truth stories. To better evaluate the stories generated by the convolutional

sequence-to-sequence model, 200-word stories are also experimented with.

5.2.2 200-Word Stories

In this task, we compare the 200-word stories generated by the above mentioned six models. However, the stories generated using Plan&Write [5], are short stories and are not a focus of this task. The story blocks for Ground_Truth, Random_Future and Random_History are truncated to 200-word stories if they are longer unlike the top three sentences used in the previous task.

The HIT experiment is similar to the previous **3-sentence Stories**, task. However, a total of four minutes are allotted for each task, with an estimated total of $0.66 per HIT instance. A separate of 100 instances are generated and tested in this experiment.

Model	Average Ranking	Standard Deviation
Ground_Truth	2.438	1.387
Random_Future	2.872	1.117
Random_History	2.906	1.345
Plan&Write	5.356	1.572
GPT2	3.230	1.488
Conv$_S eq2Seq$	4.198	1.406

Table 5.3. Human Evaluation Results for 200-word stories generated by the model

Based on the results reported in 5.3, it is observed that the model generated stories fail to outperform the baseline stories. Although, the stories generated by convolutional sequence-to-sequence model outperforms all the stories generated by the other generating models. But, it is safe to state that generation of unique and engaging long stories is still a daunting task which requires a lot more research. Also, we get the expected ranking for the baseline models which is Ground_Truth being ranked the highest followed by Random_Future and then Random_History.

5.2.3 Discussion

In this work, we see the performance of the Convolutional Sequence-to-Sequence architecture for producing follow-up arcs of human written stories. We try to ascertain if the model is capable of following the story that has been established so far and generate a follow-up passage that is inline with the story that has been written so far.

Based on the human evaluation results, we see that human readers do prefer the follow-up story arc produced by the model discussed and explored in this thesis work

over other generation models. The results for one of the testing instances in reported in Table 5.5. The story generated by Plan&Write is exceptionally short and fails to provide a lot of description and generating an engaging story. Also the generated text seems to have very abrupt endings from a literary standpoint with minuscule interconnection between the generated sentences. While the story generated by GPT also suffers to provide a colorful descriptions in the generated story failing to keep the readers engaged.

However, the model is still incapable of generating stories which match the original story line to beat the ground-truth. One of the reason for this could be that there are still certain instances where the model fails to follow the premise even with the fusion architecture. The model fails to stay on a topic and ends up generating incoherent texts as seen in Example 5.4. These fallacies in the model's generation capabilities warrants for further work to be conducted in this field of generation tasks to develop coherent follow-up story arcs.

Example 1
He stood and walked over to the refrigerator and grabbed a handful of water from the fridge. The **water** was hot but it was hot and hot. He took a small **cup of tea** and lit a cigarette. He walked back and sat on the edge of the bed. He picked up a **cup of coffee** and looked up at the sky. He was just about to start the shower when the light came to a full-moon.
Example 2
He stood and walked over to the refrigerator and grabbed a handful of water from the fridge. **The water was hot but it was hot and hot**. He took a small cup of tea and lit a cigarette. He walked back and sat on the edge of the bed. He picked up a cup of coffee

Table 5.4. Certain instances where the model generates incoherent text warranting further work. In the particular instance the model fails provide seamless transition from water to tea to coffee due to its inability to stay on a topic as seen in *Example 1*. The model also suffers from some repetition as seen in *Example 2*.

MODEL	GENERATED STORY
Ground_Truth	Diana said in utter confusion. Mel looked like she'd just been beamed into winning the lottery of entertainment material. cried Percy's friend as his girlfriend sidled up next to him.
Random_Future	And I would always smile and say, 'Sure.' "But I prepared this pamphlet. And she says, 'Raina. So I stayed.
Random_History	She walked over to the door, paused and noticed asher feet sank into a deep red Persian carpet. The trapdoor creaked open. Trapped within stood a tiny little Pegasus
Plan&Write	i was n't sure if the woman had been . she did n't want to be able to go back to me , and he did n't know what to do . i could n't even tell him about it .
GPT2	One of the men, a middle-aged man in his early thirties, spoke up. The other man, a middle-aged man in his early thirties, shook his head. The other man, a middle-aged man, spoke up.
Conv_Seq2Seq	And he had a good idea of how she was going to get him in trouble before she could get him off the hook. ... The man he'd lived with would never have been able to make such an obvious difference. He must be a man

Table 5.5. Generated Stories by models. For Seq2Seq only the first and last sentence are displayed.

Bibliography

[1] Mostafazadeh, N., N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen (2016) "A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, San Diego, California, pp. 839–849.
URL `https://www.aclweb.org/anthology/N16-1098`

[2] Mostafazadeh, N., A. Kalyanpur, L. Moon, D. Buchanan, L. Berkowitz, O. Biran, and J. Chu-Carroll (2020) "GLUCOSE: GeneraLized and COntextualized Story Explanations," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, pp. 4569–4586.
URL `https://www.aclweb.org/anthology/2020.emnlp-main.370`

[3] Fan, A., M. Lewis, and Y. Dauphin (2018) "Hierarchical Neural Story Generation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898.

[4] Zhu, Y., R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler (2015) "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books," in *The IEEE International Conference on Computer Vision (ICCV)*.

[5] Yao, L., N. Peng, R. Weischedel, K. Knight, D. Zhao, and R. Yan (2019) "Plan-and-write: Towards better automatic storytelling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7378–7385.

[6] Riedl, M. O. and R. M. Young (2010) "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, **39**, pp. 217–268.

[7] Gervás, P., B. Díaz-Agudo, F. Peinado, and R. Hervás (2005) "Story plot generation based on CBR," *Knowl. Based Syst.*, **18**, pp. 235–242.

[8] Swanson, R. and A. Gordon (2012) "Say Anything: Using Textual Case-Based Reasoning to Enable Open-Domain Interactive Storytelling," *ACM Trans. Interact. Intell. Syst.*, **2**, pp. 16:1–16:35.

[9] Liu, P. J., M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer (2018) "Generating Wikipedia by Summarizing Long Sequences," *CoRR*, **abs/1801.10198**, 1801.10198.
URL http://arxiv.org/abs/1801.10198

[10] Zhang, X. and M. Lapata (2014) "Chinese Poetry Generation with Recurrent Neural Networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, pp. 670–680.
URL https://aclanthology.org/D14-1074

[11] Roemmele, M. (2016) "Writing Stories with Help from Recurrent Neural Networks," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, AAAI Press, p. 4311–4312.

[12] Kiros, R., Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler (2015) "Skip-Thought Vectors," *CoRR*, **abs/1506.06726**, 1506.06726.
URL http://arxiv.org/abs/1506.06726

[13] Jain, P., P. Agrawal, A. Mishra, M. Sukhwani, A. Laha, and K. Sankaranarayanan (2017) "Story Generation from Sequence of Independent Short Descriptions," *CoRR*, **abs/1707.05501**, 1707.05501.
URL http://arxiv.org/abs/1707.05501

[14] Martin, L. J., P. Ammanabrolu, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl (2017) "Event Representations for Automated Story Generation with Deep Neural Nets," *CoRR*, **abs/1706.01331**, 1706.01331.
URL http://arxiv.org/abs/1706.01331

[15] Harrison, B., C. Purdy, and M. Riedl (2021) "Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, **13**(2), pp. 191–197.
URL https://ojs.aaai.org/index.php/AIIDE/article/view/13003

[16] Ammanabrolu, P., E. Tien, W. Cheung, Z. Luo, W. Ma, L. J. Martin, and M. O. Riedl (2019) "Story Realization: Expanding Plot Events into Sentences," *CoRR*, **abs/1909.03480**, 1909.03480.
URL http://arxiv.org/abs/1909.03480

[17] Fan, A., M. Lewis, and Y. Dauphin (2019) "Strategies for Structuring Story Generation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp. 2650–2660.
URL https://www.aclweb.org/anthology/P19-1254

[18] Bamman, D., B. O'Connor, and N. A. Smith (2013) "Learning Latent Personas of Film Characters," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Sofia, Bulgaria, pp. 352–361.
URL https://www.aclweb.org/anthology/P13-1035

[19] Huang, C.-Y. and T.-H. K. Huang (2021) "Semantic Frame Forecast," *arXiv preprint arXiv:2104.05604*.

[20] Zheng, H. and M. Lapata (2019) "Sentence Centrality Revisited for Unsupervised Summarization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp. 6236–6247.
URL https://www.aclweb.org/anthology/P19-1628

[21] Chaudhury, A., M. Tapaswi, S. W. Kim, and S. Fidler (2019) "The Shmoop Corpus: A Dataset of Stories with Loosely Aligned Summaries," *arXiv:1912.13082*.

[22] Jozefowicz, R., O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu (2016), "Exploring the Limits of Language Modeling," 1602.02410.

[23] Dauphin, Y. N., A. Fan, M. Auli, and D. Grangier (2017) "Language Modeling with Gated Convolutional Networks," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 933–941.
URL https://proceedings.mlr.press/v70/dauphin17a.html

[24] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin (2017) "Attention is All you Need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc.
URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[25] Gehring, J., M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin (2017), "Convolutional Sequence to Sequence Learning," 1705.03122.

[26] Sriram, A., H. Jun, S. Satheesh, and A. Coates (2017), "Cold Fusion: Training Seq2Seq Models Together with Language Models," 1708.06426.

[27] URL https://etda.libraries.psu.edu/catalog/24502kxk754

[28] Hsu, T.-Y., C.-Y. Huang, Y.-C. Hsu, and T.-H. Huang (2019) "Visual Story Post-Editing," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, pp. 6581–6586.
URL https://aclanthology.org/P19-1658