The Pennsylvania State University
The Graduate School

STORY GENERATION USING INTERMEDIATE PLOT REPRESENTATION

A Thesis in
Computer Science and Engineering
by
Kavya Laalasa Karanam

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2021

The thesis of Kavya Laalasa Karanam was reviewed and approved by the following:

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

Clyde Lee Giles
Professor of Computer Science and Engineering
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

Writing the stories had always been a challenging task as it is always possible to get confused when writing huge drafts with limited constraints. Automated story telling is process where AI is used to structure and generate long stories when certain lines of a story is given as an input. Even for human beings it is really a struggle to craft or come up with good stories especially long connected sentences which are meaningful . There have been numerous datasets that were previously experimented or tested in this creative task. However, most of datasets comprises simple and short paragraphs which contains approximately 5 to 7 sentences that restricts complexity that the machines have to deal with to learn the stories making the story generation task straightforward. This work investigates that for a given set of story blocks, if story generation models would be able to construct practical follow-up stories for realistic human-written long stories. All the experiments are performed on BookCorpus dataset which proved that the model is better at generating short summaries than longer summaries for the generated follow-up stories. Human Evaluation results indicate that stories are better ranked for generated summaries than the human-written ground truth stories.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I am gratefully indebted to my advisor Dr. Ting-Hao (Kenneth) Huang for his continuous trust, patience, suggestions, time and help during my Master's Thesis Journey. He consistently provided me the guidance to make sure that I was making significant progress in the project.

I am thankful to Dr. C. Lee Giles for accepting my request to be the advisor and his input, feedback & suggestions on my research.

I am thankful to Dr. Rui Zhang for agreeing to be in my thesis committee and for his constant support and valuable insights for the project.

I am thankful to my co-advisor Dr. Arzoo Katiyar for her unfailing support, valuable insights and continuous encouragement for the project implementation.

I would like to thank Chieh-Yang Huang for being a great mentor who inspired me and helped me gain a broader perspective in my research. I feel blessed to have known and spent time with my friends, roommates and seniors outside my major and have learnt a lot from them and received so much love from people here. Thank you Sumedha Prathipati, Sneha Galiveeti, Shivani Chopra, Pranitha Malae, Palaniappan Meenaakshi Sundaram, Anerudh Vijayaraghavan, Tarun Sathapathi, Saniya Naphade, Murali Nandan Nagarapu, Sahithi Rampalli and Nagadastagiri Challapalle for making all those challenging days sustainable and memories cherishable.

Words are inadequate to express my gratitude to my parents - Prem Kumar and Usha Rani, my grandparents - Selvarathnam and Seshamma for their understanding, encouragement, unconditional love and unfailing support in my entire Master's Journey.

Dedication

I dedicate this thesis to my parents and my grandparents.

Chapter 1
Introduction

1.1 Story generation

Storytelling as a creative task can be described as the art of passing or conveying information among human beings. Writing a story is extremely hard because it is often difficult to come up with set of sentences which are connected depending on various factors like plot, actors and building features based on one's imagination. Every writer encounters various challenges to come up with such stories which are coherent as well us diverse in nature. There are various research areas and industry proven techniques like supporting sentence level constructs or structures like auto-completion, basic grammatical error detection mechanisms or spell checkers that aid this storytelling task. However, for the high end tasks like the story generation, it was always treated as reckoning deep learning or machine learning problem.

Various story datasets comprising short stories like five sentences long such as ROC-Stories [1] and GLUCOSE [2] where in each each story specific causal statement is included with an inference rule was developed with required constraints which eases the machine learning model to learn quickly. Some other works like WritingPrompt [3] have included a starter prompt with every story in the dataset to generate these stories relevant to these starter prompts. In real world story telling tasks, it is a tedious tasks as the novels are generally composed with more than 1,00,000 words consisting of long sentences and working on these limited constraints to construct stories would make this creative task not only cumbersome but unrealistic. Existing works illustrated above generate the stories acquiring the knowledge from the dataset when they are trained, it is uncertain how these tasks aid the real world story tellers.

As part of this work, Story generation models are determined of their abilities to generate the follow-up stories for a given set of human-written story sentences. A pre-arranged set of stories consisting of 20 sentences are considered as story blocks which is considered as a long novel is taken as the input data and determine if the story generation model can construct a story for the next story block i.e, S_B_{n+1} where previous story blocks of 20 sentences long are given i,e, $\{S\_B_n, S\_B_{n-1}, S\_B_{n-2}, \cdots\}$. This conceptualization is indistinguishably close to the semantic frame forecast task [4], where it determines the representation of S_B_{n+1}. Generating these stories with a block size comprising of 20 sentences is cumbersome task as the sentences are really long considering the real world dataset Bookcorpus [5]. Therefore, this work aims to generate a summary than the full sentences story for a given target story block S_B_{n+1}.

This work involves experimenting Plan-and-Write [6] where training different plot planning models on automatically extracted storylines on the story summary blocks produces better next story summary block without needing supplementary human annotations. Both the dynamic planning and static planning are experimented for this tasks to understand how these frameworks improve the generated story summary blocks in terms of both coherence and diversity aspects. This plan-and-write framework are aimed to generate stories from given title of the story. We experiment this framework to understand if this can be extended to generate the next target story summary block given the current and previous story summary blocks as input.

All the experiments conducted on this real world dataset Bookcorpus [5] proves that the neural story generation models such as Plan-and-Write [6] performs well on semantic based metrics than Fusion-based seq2seq [3] which performs better in word overlapping based metrics than GPT-2 [7] Model which is good at producing short (*e.g.,* 3-sentence) summaries for the follow-up story blocks in automatic evaluation. However, for human evaluation crowd workers ranked the generated story summaries which are short better than the directly target human written gold stories. This work involves analyzing the challenges involved in story generation tasks, experimenting with Plan-and-Write [6] on the Book Corpus Dataset, evaluating the generated short story summaries on how it will be helpful for story writers.

Chapter 2
Related Work

Early research on story generation task involved computational narrative reasoning to create event sequences which are connected and meaningful. Many factors like appreciation of beautifully crafted stories which are logically connected along with comprehensibility attributes to these narratives. These narratives are dependent on factors like plot progression and how the characters closely act according to the audience expectations. This work Narrative Planning [8] uses Novel refinement search planning algorithm called the Intent-based Partial Order Causal Link (IPOCL) planner which generates better narrative plots which supports good conception for character intentions. Another interesting work Computational Model of Plan-Based Narrative Conflict at the Fabula Level [9] in narrative planning is presence of conflict. This work ensured that there exists narrative conflict in a plan while sustaining character intentions and better narrative causal link planning by defining conflict in 7 dimensions as a combination of both discrete and continuous values targeting knowledge representations. This work claims to generate stories using conflict partial order causal link-planning algorithm (CPOCL) based on the limited constraints imposed across the 7 different dimensions.

Different works have proven to automatically construct the story plots for entertainment industry but compromising on the quality of these plots generated given the initial conditions of the storyline and also the characters involved in the story. However these generated plots needed additional effort like selection and alteration from various screen writers for validation and reliability. One of the works - Story Plot Generation based on CBR [10] designed the plot structure using case based reasoning on the entire dataset of stories to generate a new story based on the given user input. Moreover, there has been several problems using case based reasoning for story telling determining what has happened so far under the same given conditions and ensuring that there exists large

case description for all the possible routes in the story. This work Say Anything: Using Textual Case-Based Reasoning to EnableOpen-Domain Interactive Storytelling [11] uses textual case-based reasoning approach for automatic interactive storytelling where both human beings and machines take alternative turns to craft a story. This system involves using dataset based on real world stories written from internet weblogs.

There has been significant improvement in the story generation research area using various neural language generation models like Towards Controllable Story Generation [12] which focuses on using various factors like ending valence(story ending - either happy or sad) and storyline keywords on both the analyzer to extract from given stories and next to supply these measures to a generator to generate the stories using rule based extraction mechanisms and conditional Recurrent Neural Networks to ensure tractability. Moreover another work explores story generation task [3] as making coherent and continuous paragraphs of sentences on any given topic. The dataset that has been used is 300000 stories written by human beings combined with prompts for these stories from an online platform. These are utilized in this story generation task in hierarchical architecture where premise is generated as the first step, followed by the transformation into full text for the stories. The architecture is further improved by adding self-attention mechanism for the capturing context information in long sentences. This architecture has shown great results by improving on baseline scores on both automatic and human evaluation results that the generated stories are ranked higher than other models developed.

There have been numerous attempts to increase the diversity and coherence of the stories generated especially researchers have tried adopt some sort of intermediate representation to help ease the process of story generation tasks. One of the foremost benefits of incorporating an intermediate representation is surface realization. One of the previous works in this area is Event Representations for Automated Story Generation with Deep Neural Nets [13] which includes an associate level of abstraction among sentences and words to preserve semantic information by controlling event sparsity. This architecture can be described as a two step process - event to event generation followed by event to sentence generation. Another improvement on this area is to produce semantically correct and related sentences which is missing in the previous architecture where it was plausible to produce grammatically correct sentences but not the later. The work Story Realization: Expanding Plot Events into Sentences [14] uses ensemble-based model to address these issues thereby generating coherent and semantically connected sentences in

the story. Further improvements on Planning and Writing long stories lead to the work Strategies for Structuring Story Generation [15] where predicate-argument structure of the sentence is first constructed followed by surface realization. This architecture has improved the consistency of all the entities and events in generated stories. Lack of cohesion on the higher level can be solved to address the problems for good plot generation to aid the story generation task. One of the works - Content Planning for Neural Story Generation with Aristotelian [16] uses a plot generation language model along with combination of multiple rescoring models to generate higher quality stories than modern baselines. However, most of these work are experimented on short stories such as WritingPrompt [3], ROCStories [1], or WikiPlots [17].

Generating pragmatic set of long sentences that describe a sequence of connected events is one of the prime reasons why automatic storytelling is demanding. Although these works in the past show significant improvement in story generation tasks, it is restricted to short stories thus narrowing down the diversity and coherence of the stories generated especially when long sentences are involved. Earlier works on plot planning included plot graphs [18] as the intermediate story line representation but this plot graphs demanded intricate specialized knowledge and is cumbersome process. Another work [19] uses a sequence of words to aid poetry composition task. There are numerous advantages of using an intermediate representation to guide the story generation process. One such works Plan-and-Write [6]is the inspiration for our work where simple curated storylines are extracted from the existing stories and use them for plot planning. Once the plots are trained - One can generate coherent and diverse stories which wont require further human annotation. This Plan-and-write explores two strategies one being dynamic planning and the other being static planning. We extend their work to check if its feasible on long stories(real world dataset) summaries i.e, to continue the story generated so far and conduct novel analysis to understand different aspects in automatic and human evaluation.

Chapter 3
Dataset Analysis and Preprocessing

3.1 Introduction

The dataset that is experimented is Bookcorpus [5] which is real world novel dataset where a total of $15,605$ raw fiction books information is curated. Several rules are incorporated to extract the necessary information from the these books whose rules are based upon the work illustrated in [4] which are as follows:

- short story books with size below 10 Kilo Bytes
- books possessing HTML related code
- books containing e-book file formats
- books containing non english stories
- "Non-Fiction", "Anthologies" and "Graphic Novels & Comics" genre books are removed.

Another important part of dataset analysis is that this data should not contain any non fictional information - regular expressions are used to remove the chapters that contain these non fictional information. All the information before the start of the first chapter including the title of the chapter and after the title of the last chapter in the book are excluded carefully. All the data after thorough cleaning a total of 4794 qualified books fiction information is extracted and split into train, validation and test sets in the following ratios of 0.70, 0.10 and 0.20 respectively. These can be summarized as training dataset of 3357 books, validation dataset of 479 books and testing dataset of 958 books

respectively. Further, these separate books are split into sequences of story blocks with each size of 20 sentences obtaining more than 900000 training instances. For testing set, randomly one story block for each book is selected to reduce the profuse nature of the dataset curated resulting in 200 testing instances for the human evaluation and 958 testing instances for the automatic evaluation.

Table 3.1. Dataset Information and Split

	Number of instances	
Total	*Books*	*Sample Sentences*
Training	3357	921421
Validation	479	1000
Testing	958	958

3.1.1 Summary Generation

The work illustrated in Pacsum [20], uses an unsupervised leaning method to produce an extractive summarization model. This Pacsum is extended to the story generation task where score weighting procedure is fine tuned using Shoomp corpus [21] which includes comprehensive stories with their summaries for each and every chapters. Empirically the curation of the non successive conversation elements reduces the output results thus, this process was modified by including a penalty score for all the conversation sentences and exclude all the descriptive sentences in the stories. For each story block, only a three-sentence plot summary is picked from the block based on the resultant scores after adding the penalty scores.

3.2 Problem Formulation

The Semantic Formulation explained in the work [4] is used in the formation of story blocks formulation of story blocks. Once a story block is generated after summarization with a fixed number of sentences in this case 3, the goal of this problem is to use a sequence of these story blocks $\{S\_B_1, S\_B_2, \cdots, S\_B_{n-1}, S\_B_n, S\_B_{n+1}, ...\}$, the task can be described as to predict the story plot of block $n+1$ using all the information before n-th story block S_B_n. Since extractive summarization idea is used - the story

summary of S_B_{n+1} is predicted using the ongoing information about the story that exists already in $\{S\_B_n, S\_B_{n-1}, ...\}$.

The idea from Plan-and-Write [6] is adapted to explore this problem statement. The work in Plan-and-Write expects the data in the format as explained below for both its static and dynamic planning models where originally the work on a given title first plans the storylines and then generate the whole story. So adapt it to this setup - the summary of $\{S\_B_n\}$ is taken as the title. For each of the 3 sentences in the summary of $\{S\_B_{n+1}\}$ is utilized to create the storyline using Rapid Keyword Extraction Algorithm(RAKE) [22].

- **Input** - A title t = $\{t_1, t_2, \cdots, t_m\}$ is taken from story summary for block $\{S\_B_n\}$ where t_i denotes the i^{th} sentence of the story summary.
- **Output** - Generate the story summary for block $\{S\_B_{n+1}\}$ as $\{s_1, s_2, \cdots, s_m\}$ where s_i denotes the i^{th} sentence of the story summary.
- **Storyline** - Model generates a storyline l= $\{l_1, l_2, \cdots, l_m\}$ for the story summary block $\{S\_B_{n+1}\}$ as an intermediate representation to further generate the sentences of the story.

3.3 Tokenization

One of the preprocessing step in any natural language generation task is tokenization. Tokenization is the process of dividing the long sentences into smaller chunks called tokens which is helpful to track patterns among words for further steps like lemmatization and handling Out of Vocabulary(OOV) constraints. Several constraints are taken into consideration while tokenizing the sentences. Natural Language Tool Kit(NLTK) Library Provides an important module called NLTK Tokenize which gives as predefined functions like word_tokenize() and sentences_tokenize() to split the large databases of texts into words and sentences. Function word_tokenize() from this library is used to split these long sentences into keywords. Further, punctuation is removed from this dataset followed by removing all the conversation elements in the texts if anything is leftover after the pacsum summarization task. Special Tokens like <EOT> - End of Title, <EOL> End of Line and </s> are made sure that they are not excluded in the process as the training data is expected to have this tokens as delimiters. Due to large number of tokens generated in the data, the model was not able to train due to huge memory constraints. Using these inbuilt tokenization function and also restricting the vocabulary size based on term frequency from 5 Million to 50000 gradually reduced the OOV Ratio from 0.79

to 0.006. Also, all the OOV words are replaced with <UNK> token to handle all these unnecessary words in the vocabulary.

3.4 RAKE Algorithm

RAKE Algorithm [22] is utilized to extract the most important keyword from the story sentence of the story block summary to construct the storyline. In any information retrieval systems, the process of extracting the necessary information is a cumbersome task. To ease this process RAKE was developed under the observation that most document systems used for information retrieval do not contain function keywords, stopwords, punctuation and abstract words which do not have proper lexical meaning as they do not help users for information search. RAKE algorithm takes the list of stop words as an input, a set of sentence delimiters and word delimiters to split the content into keywords and scores are assigned based on co-occurrence factor and association among these keywords. Keyword extraction is the next step where the content and skims through the set of generated keywords by first splitting the sentences using sentence delimiter, followed by words array by word delimiters and excluding stop words. Based on several evaluation metrics from the degrees of the graphs constructed from these keywords and any of their scores summed up together like deg(w), freq(w), deg(w)/freq(w). RAKE internally handles interior stop words by finding adjacent pair of words that occur multiple times in the whole content. Finally Top(1/3) scored keywords from the entire document are selected as candidate keywords. RAKE Algorithm has proven to be efficient to extract the most important keyword when compared to other competetive baselines TextRank and Ngram with Tag metrics. Utilizing the power of Natural Language Tool Kit Library - RAKE Algorithm was implemented which was used in our work to extract the story line keywords.

Chapter 4
Experiments

4.1 Methods

4.1.1 Introduction

To adopt the models described in the paper [6] - there are two architectures experimented with the given dataset - **Dynamic Planning** and **Static Planning** where each of them can be briefly distinguished as one being content-introducing problem and the other being modelled as conditional generation problem respectively.

4.1.2 Dynamic Planning - Bidirectional GRU

One of the biggest benefits of this architecture is it malleability. In each step the next word of the storyline is predicted followed by the next sentence of the story summary block. All the steps to generate the story carefully captures information at each instance from the current storyline generated so far combining with all the previously generated story sentences.

4.1.2.1 Storyline Planning

At each step as illustrated before storyline is first planned based on the context information generated so far which is the previously generated sentences combined with the title of the story(sentences from the block S_B_n) and the previously generated storyline word at that instance. This can be termed as content-introducing problem where in each step a new word in the storyline is produced based on the contextual information captured till that step combined with recently generated word from the storyline. The context can be explained mathematically as -

$$context = [S\_B_n, S\_B_{n+1}[1 : i-1]$$

where S_B_n can be modelled as the sentences of story block taken as title and the $S\_B_{n+1}[1 : i-1]$ as the first i-1 sentences generated for the story block S_B_{n+1}. The probability $Prob(l_i|context, l_{i-1}; \theta)$ is calculated as per the work described in Towards implicit content-introducing for generative short-text conversation systems [23] where Bi-Directional gated recurrent unit(Bi-GRU) is used to encode the context information and then the additional information is appended to it which is l_{i-1} to the decoder after concatenating the hidden vectors element wise.

4.1.2.2 Story Generation

Incrementally the storyline is planned followed by the story sentences generated in the following dynamic planning architecture. This story generation process can be depicted as another content introducing problem where the sentence of the story summary block S_B_{n+1} is generated based where in each step a new sentence in the story is produced based on the contextual information captured till that step combined with recently generated sentence from the story. Model is trained end-to-end such that the log likelihood of the probability of the training data is minimized. [6]

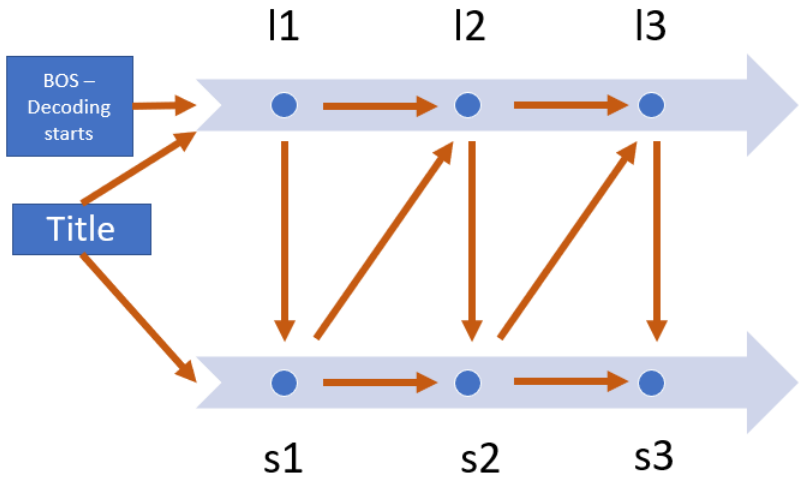

Figure 4.1. Dynamic Planning Architecture

4.1.3 Static Planning - Bidirectional LSTM

The work presented in Plan-and-write [6] clarifies that Static Planning does give coherent and diverse stories even though it compromises on the malleability factor but it does help by giving some insight into the future by completing the storyline or plot generation prior to generating the first sentence of the story.

4.1.3.1 Storyline Planning

At each step as illustrated before storyline is first planned completely based on the context information which is just the title of the story so in our case sentences from the block S_B_n. This can be termed as conditional-generation problem where all the words in the title is encoded using Bi-directional LSTM architecture which further on decoding using a unidirectional LSTM architecture generates the whole plot or storyline for the block S_B_{n+1}. The probability $Prob(l_i|t, l_{1:i-1}; \theta)$ is calculated as per the work

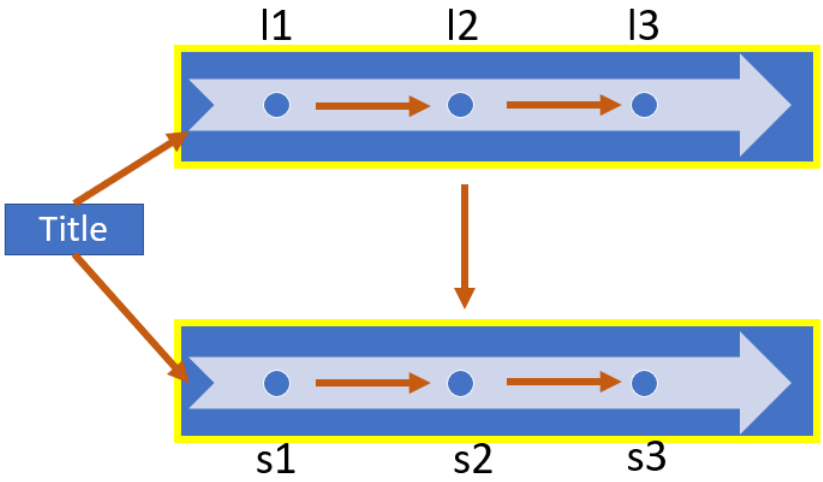

Figure 4.2. Static Planning Architecture

described in Neural Machine Translation by Jointly Learning to Align and Translate [24] where Bi-Directional Long short term memory network(Bi-LSTM) is used to encode the the title. The conditional probability can be explained mathematically as -

$$Prob(l_i|t, l_{1:i-1}; \theta) = Multi\ Layer\ Perceptron(Encode(S\_B_n), l_{i-1}, decoding\ hidden\ state_{i-1})$$

where S_B_n can be modelled as the sentences of story block taken as title and the $l_{1:i-1}$ as the first i-1 keywords in storyline generated for the story block S_B_{n+1}.

4.1.3.2 Story Generation

The sentences of the story summary block S_B_{n+1} is generated after the complete storyline is produced, That is the story summary block generation is a different conditional generation problem where a new Bi-Directional Long short term memory network is trained where it is responsible for encoding both the title(Sentences of story summary block S_B_n) and the storyline generated for the story summary block S_B_{n+1} with '<EOT>' token. Model is trained end-to-end such that the log likelihood of the probability of the training data is minimized. [6].

4.1.4 Storyline Optimization and why Static Planning?

To illustrate how this architecture can be extended to the dataset and our problem statement - we can depict the whole process of story generation as per the flowchart.

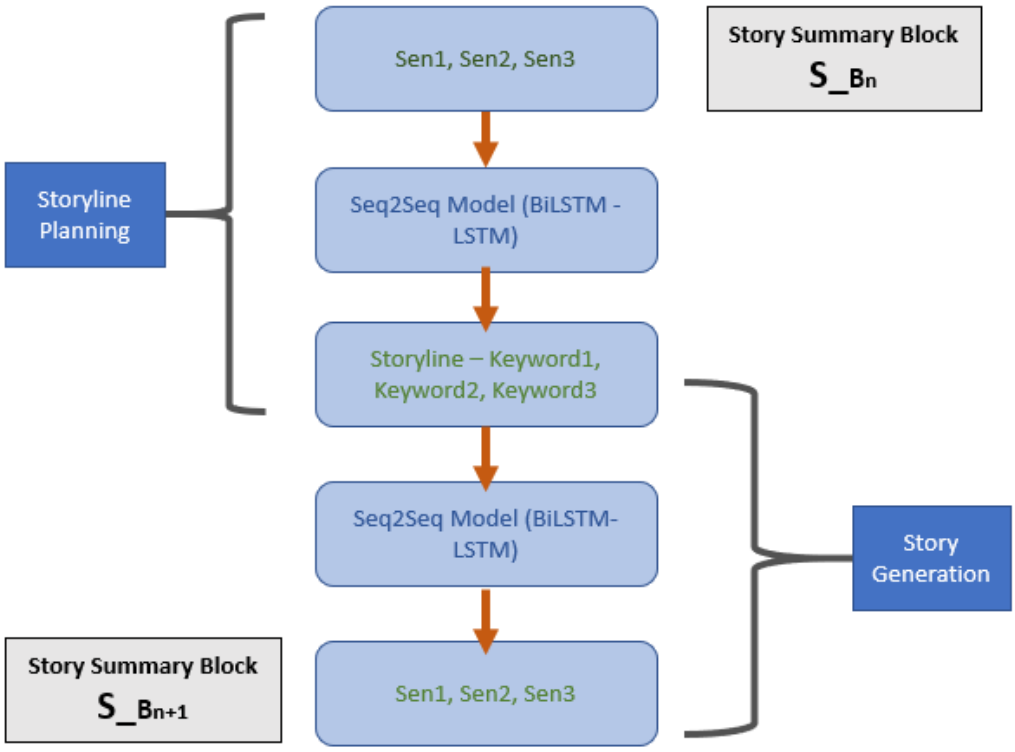

Figure 4.3. Flowchart

Among all the problems exhibited by the neural language generation models repetition among the words or sentences is peculiar. To prevent this repetition particularly in story sentences generated this architecture gives a feasibility of modifying the generated storyline before the story generation by applying these rules.

Chapter 5
Discussion and Results

5.1 Automatic Evaluation

On all the 956 instances of the testing set are automatically evaluated on the stories generated from the static planning model against the target gold stories that the dataset possess. The work Plan-and-write [6] illustrates that the experimented results conducted from on ROCStories [1] gave better stories with respect to several metrics like coherence, fidelity, inter-story separation and intra-story separation. Therefore, this Static planning architecture results are evaluated where a story summary block of 20 sentences is taken as the gold standard for the target stories. NLG-eval package [25] is extended and adopted to generate these four metrics : BLEU-4, METEOR, ROUGE-L and SkipThought Cosine Similarity (ST-CS) scores are reported in the following Table 5.1. The results from other

Table 5.1. Automatic Story Evaluation for the entire story block as reference where BLEU-4, METEOR and ROUGE-L are word overlapping metrics and ST-CS being the semantic based metric

	Model				
Metrics	*Plan&write*	*Fusion-Seq*	*GPT2*	*RandomFuture*	*RandomHistory*
BLEU-4	0.0000	0.0026	0.0002	0.0002	0.0001
METEOR	0.0315	0.1014	0.0352	0.0306	0.0316
ROUGE-L	0.0732	0.1414	0.0864	0.0739	0.0763
ST-CS	0.5586	0.3436	0.5348	0.5525	0.5595

baselines like Fusion-Seq, GPT2 are better in word-overlapping metrics however the static planning architecture gives decent semantic based metric. The work Plan-and-Write [6] states that the story generation task the word-overlapping based metrics like BLEU-4 is

not suitable especially because the target stories can be different and equally good when compared to the generated stories [26].

5.2 Human Evaluation

Human evaluation is conducted using Amazon Mechanical Ture(Mturk). Several experiments are conducted to evaluate how the model performs especially when dealing with long sentences.

- 3-sentence story summary
- 200-word story summary

5.2.0.1 3-sentence story summary

In this experiment three baselines are included for comparison out of which Random-Future is the strongest baseline and Random-History being the weak baseline.

- Ground-Truth : story summary on S_B_{n+1} which is expected to be the upper bound
- Random-Future : story summary on a randomly selected story block from S_B_{n+5} to S_B_{n+15}
- Random-History: story summary on a randomly selected story block before S_B_{n-10}

Two other models results which is collaborators work are used to compare the results but it is excluded from this work as it is out of scope.

- Fusion-Seq : story summary on S_B_{n+1} for which story block S_B_n is presented as the input using Seq2Seq Model [3].
- GPT2 : story summary on S_B_{n+1} for which story block S_B_n is presented as the input using Semantic Frame Representations [4].

In this Human Intelligence Task (HIT) all the mechanical workers are given both the story block S_B_n and 3-sentence story generated for story block S_B_{n+1} by Plan&Write and other two baselines to rank the plot ideas from rank 1 to rank 6 with rank 1 being the highest focusing on the quality of the plots and excluding minimalistic issues repetition

Table 5.2. 3-Sentence Human Evaluation Result

	Metrics	
Model	*Average Ranking*	*Standard Deviation*
Plan&Write	4.372	1.624
Ground-Truth	3.432	1.646
Random-Future	3.620	1.552
Random-History	3.564	1.544
Fusion-Seq	2.566	1.831
GPT2	3.446	1.543

and connotations. A definite time is enforced on these workers like 30-seconds to make sure that the workers read the stories before ranking them. All the different workers are explicitly required to have these main qualifications of possessing more than 98% Approval Rate with more than 3000 Approved HITs with Adult Content Qualification. Each of these HIT takes minimum of than two minutes to complete approximately resulting in price of $0.33 per HIT (Hourly wage = $10/hr).

Table 5.3. 200-Word Story Summary Evaluation Result

	Metrics	
Model	*Average Ranking*	*Standard Deviation*
Plan&Write	5.356	1.572
Ground-Truth	2.438	1.387
Random-Future	2.872	1.117
Random-History	2.906	1.345
Fusion-Seq	3.230	1.488
GPT2	4.198	1.406

A total of 100 instances are evaluated under this experiment. The results for the following experiment are shown in the Table 5.2. Among all the 3-sentence story generation baselines workers prefer the ground truth, Random-Future and Random-History more than the Plan-and-Write Architecture. This is mainly because of longer sentences in the other baselines than leaving ground truth at a better rank than random history because summarization Model cannot work well with real world novel fiction and cannot clearly specify the meaning of the whole story block accurately.

5.2.0.2 200-word story summary

This experiment is comparing different models when the stories are generated with 150-200 words. We compare models when generating stories with 150-200 words. As the Plan & Write Architecture involves generating the storyline as the prior step thus the same story generated is uses as the 200-word story summary in all the baselines like Ground-Truth, Random-Future, and Random-History includes story summaries generated until 150 words instead of the top three sentences. The Human Intelligence Task(HIT) is same as the previous experiment but workers were able to finish the task in four minutes empirically at price of $0.66 per HIT. Totally, 100 instances are generated and used for this experiment to prevent workers from rating the same story. The result is reported in the Table 5.3. When generating long stories, Plan & Write does not perform well when compared to the baselines proving that it doesn not go well with fictional stories containing long sentences. We can conclude that generating coherent, diverse and interesting **long** stories is still cumbersome task. The Ranking of all the baselines from above scores can be describes as increasing order of performance - Plan & Write, Ground-Truth, Random-Future and Random-History. This proves from our previous assertion that any summarization based deep learning model may not work well for generating fictional stories summaries thus concluding that presence of longer sentences does make more meaningful stories.

5.3 Discussion

We understand that the main motive of this work is to understand if Plan and Write architecture supports the story writers by suggesting follow up arcs. The metrics taken for automatic evaluation and human evaluation has certain limitations. For the human evaluation, the five workers are made sure that they rank the 6 different models from 1 to 6 with 1 being the highest in terms of quality, connotation and repetition of sentences. But these results are not concrete in terms of ranking as we do not understand which model lacks which of these areas separately. Therefore, incorporating these metrics where each of the models are evaluated and ranked individually in terms of repetition of sentences, quality and connotation would give better insights on how can a model be further improved based on where it lacks. Further, TLDR - Extreme Summarization for Scientific documents [27] can be adapted to further improve the story generation process where both the story summary block generation along with the storyline generation can be generated similar to how TLDRs and titles are generated using Controlled

Abstraction for TLDRs with Title Scaffolding(CATTS). Another inspiring work can also be used to extend this story generation task - D2S: Document-to-Slide Generation Via Query-Based Text Summarization [28] where all the relevant text, figures and text are retrieved using the slides and summarize them into important points. This two step process can be adapted for the story summary generation task. Another exciting work, Extractive Research Slide Generation Using Windowed Labeling Ranking [29] - where SummaRuNNer a sequence model for extractive summarization task ranks all the sentences based on how closely they are semantically related after combining the lexical features obtained within a definite window size. All these slide generation modules takes the advantages of the document structures while generating the slides. Example all the scientific documents follows introduction, related work, methods, experiments and results which can be utilized when generating the slides. However, this Similar Knowledge from work can be incorporated where the structures of the stories [30] can be taken into account for the stories to improve our story summary block generation performance.

Table 5.4. Output Stories generated by the Models

	Results
Model	*3- Sentence Stories generated*
Plan&Write	i caught my breath . i could see the pain on my face . he had n't been able to play up , and i was going to get it off .
Ground Truth	Once inside she examined the room. Katie walked closer and found what she was looking for. She whispered to herself, and reached out for it, and again she was hesitant.
Random - Future	She walked over to the door, paused and noticed as her feet sank into a deep red Persian carpet. The trapdoor creaked open. Trapped within stood a tiny little Pegasus.
Random - History	Galaxy's mane flew up and down, the little pink flowers still braided in her long wavy mane. Then suddenly Katie felt a tickling feeling on her right leg. She looked down surprised and then noticed a small foal, a yearling galloping next to Galaxy.
Fusion-Seq	A few more minutes before she got the hang of it and sat down on the edge of the bed. Katie looked over at the clock. The clock was already ticking in the morning, and Katie was already on the bed when Katie got up from the bed. · · · She was n't sure how she would react to that when she was in the room and that she had no idea if she could be anywhere
GPT2	Katie asked. There was a knock at the door. The door opened and Heath came in, followed by Heath and Gertrude, carrying cups of steaming herbal tea and two mugs of steaming hot tea that were steaming together

Chapter 6
Conclusion

Existing story generation models on real world datasets like BookCorpus [5] are applied to understand if it can support writers by suggesting the follow up arcs. The experiments reveal that generated story summaries achieve lower ranking in human evaluation for Plan & Write [6] Model than other baselines. For generating long stories, summarization model is not good enough for generating fiction summaries as long sentences are required. Moreover, all the generated stories from Plan & Write Model is fairly simple when compared to other baselines. Therefore, we can conclude that generated story summaries from Plan & Write architecture is average when compared to target human-written gold stories but it suffers on real fictions.

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

[5] 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)*.

[6] 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.

[7] Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019) "Language models are unsupervised multitask learners," *OpenAI blog*, **1**(8), p. 9.

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

[9] Ware, S. G., R. M. Young, B. Harrison, and D. L. Roberts (2013) "A computational model of plan-based narrative conflict at the fabula level," *IEEE Transactions on Computational Intelligence and AI in Games*, **6**(3), pp. 271–288.

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

[11] 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.

[12] Peng, N., M. Ghazvininejad, J. May, and K. Knight (2018) "Towards Controllable Story Generation," in *Proceedings of the First Workshop on Storytelling*, Association for Computational Linguistics, New Orleans, Louisiana, pp. 43–49.
URL https://www.aclweb.org/anthology/W18-1505

[13] Martin, L. J., P. Ammanabrolu, X. Wang, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl (2018) "Event representations for automated story generation with deep neural nets," in *Thirty-Second AAAI Conference on Artificial Intelligence*.

[14] Ammanabrolu, P., E. Tien, W. Cheung, Z. Luo, W. Ma, L. J. Martin, and M. O. Riedl (2020) "Story Realization: Expanding Plot Events into Sentences." in *AAAI*, pp. 7375–7382.

[15] 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

[16] Goldfarb-Tarrant, S., T. Chakrabarty, R. Weischedel, and N. Peng (2020), "Content Planning for Neural Story Generation with Aristotelian Rescoring," 2009.09870.

[17] 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

[18] Li, B., S. Lee-Urban, G. Johnston, and M. Riedl (2013) "Story generation with crowdsourced plot graphs," *Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013*, pp. 598–604.

[19] Wang, Z., W. He, H. Wu, H. Wu, W. Li, H. Wang, and E. Chen (2016), "Chinese Poetry Generation with Planning based Neural Network," 1610.09889.

[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] Rose, S., D. Engel, N. Cramer, and W. Cowley (2010) *Automatic Keyword Extraction from Individual Documents*, pp. 1 – 20.

[23] Yao, L., Y. Zhang, Y. Feng, D. Zhao, and R. Yan (2017) "Towards Implicit Content-Introducing for Generative Short-Text Conversation Systems," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Copenhagen, Denmark, pp. 2190–2199.
URL https://www.aclweb.org/anthology/D17-1233

[24] Bahdanau, D., K. Cho, and Y. Bengio (2016), "Neural Machine Translation by Jointly Learning to Align and Translate," 1409.0473.

[25] Sharma, S., L. El Asri, H. Schulz, and J. Zumer (2017) "Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation," *CoRR*, **abs/1706.09799**.
URL http://arxiv.org/abs/1706.09799

[26] 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*, pp. 6581–6586.

[27] Cachola, I., K. Lo, A. Cohan, and D. S. Weld (2020), "TLDR: Extreme Summarization of Scientific Documents," 2004.15011.

[28] Sun, E., Y. Hou, D. Wang, Y. Zhang, and N. X. R. Wang (2021), "D2S: Document-to-Slide Generation Via Query-Based Text Summarization," 2105.03664.

[29] Sefid, A., J. Wu, P. Mitra, and L. Giles (2021), "Extractive Research Slide Generation Using Windowed Labeling Ranking," 2106.03246.

[30] Snyder and Blake (2005), "Save the Cat! : the Last Book on Screenwriting You'll Ever Need." .