Plan-And-Write: Towards Better Automatic Storytelling

Lili Yao[1,3*], **Nanyun Peng**[2*], **Ralph Weischedel**[2], **Kevin Knight**[2], **Dongyan Zhao**[1] **and Rui Yan**[1†]
liliyao@tencent.com, {npeng,weisched,knight}@isi.edu
{zhaodongyan,ruiyan}@pku.edu.cn
[1] Institute of Computer Science and Technology, Peking University
[2] Information Sciences Institute, University of Southern California, [3] Tencent AI Lab

Abstract

Automatic storytelling is challenging since it requires generating long, coherent natural language to describes a sensible sequence of events. Despite considerable efforts on automatic story generation in the past, prior work either is restricted in plot planning, or can only generate stories in a narrow domain. In this paper, we explore open-domain story generation that writes stories given a title (topic) as input. We propose a *plan-and-write* hierarchical generation framework that first plans a storyline, and then generates a story based on the storyline. We compare two planning strategies. The *dynamic* schema interweaves story planning and its surface realization in text, while the *static* schema plans out the entire storyline before generating stories. Experiments show that with explicit storyline planning, the generated stories are more diverse, coherent, and on topic than those generated without creating a full plan, according to both automatic and human evaluations.

Introduction

A narrative or story is anything which is told in the form of a causally/logically linked set of events involving some shared characters (Mostafazadeh et al. 2016a). Automatic storytelling requires composing coherent natural language texts that describe a sensible sequence of events. This seems much harder than text generation where a plan or knowledge fragment already exists. Thus, story generation seems an ideal testbed for advances in general AI. Prior research on story generation mostly focused on automatically composing a sequence of events that can be told as a story by plot planning (Lebowitz 1987; Perez and Sharples 2001; Porteous and Cavazza 2009; Riedl and Young 2010; Li et al. 2013) or case-based reasoning (Turner 1994; Gervas et al. 2005). These approaches rely heavily on human annotation and/or are restricted to limited domains. Moreover, most prior work is restricted to the abstract story representation level without surface realization in natural language.

In this paper, we study generating natural language stories from any given title (topic). Inspired by prior work on dialog planning (Nayak et al. 2017) and narrative planning (Riedl and Young 2010), we propose to decompose story generation into two steps: 1) story planning which generates plots, and 2) surface realization which composes natural language text based on the plots. We propose a *plan-and-write* hierarchical generation framework that combines plot planning and surface realization to generate stories from titles.

Title (Given)	The Bike Accident
Storyline (Extracted)	Carrie → bike → sneak → nervous → leg
Story (Human Written)	Carrie had just learned how to ride a bike. She didn't have a bike of her own. Carrie would sneak rides on her sister's bike. She got nervous on a hill and crashed into a wall. The bike frame bent and Carrie got a deep gash on her leg.

Table 1: An example of title, storyline and story in our system. A storyline is represented by an ordered list of words.

One major challenge for our framework is how to represent and obtain annotations for story plots so that a reasonable generative model can be trained to plan story plots. Li et al. [2013] introduces plot graphs which contain events and their relations to represent a storyline. Plot graphs are comprehensive representations of story plots, however, the definition and curation of such plot graphs require highly specialized knowledge and significant human effort. On the other hand, in poetry composition, Wang et al. [2016] provides a sequence of words to guide poetry generation. In conversational systems, Mou et al. [2016] takes keywords as the main gist of the reply to guide response generation. We take a similar approach to represent a story plot with a sequence of words. Specifically, we use the order that the words appear in the story to approximate a storyline. Table 1 shows an example of the title, storyline, and story.

Though this representation seems to over-simplify story plots, it has several advantages. First, because the storyline representation is simple, there are many reliable tools to extract high-quality storylines from existing stories and thus automatically generate training data for the plot planning model. Our experiments show that by training plot planning models on automatically extracted storylines, we

[*] Equal contribution: Lili Yao and Nanyun Peng

[†] Corresponding author: Rui Yan (ruiyan@pku.edu.cn)

can generate better stories without additional human annotation. Moreover, with this simple and interpretable storyline representation, it is possible to compare the efficiency of different plan-and-write strategies. Specifically, we explore two paradigms that seem to mimic human practice in real world story writing[1] (Alarcon 2010). The *dynamic* schema adjusts the plot improvisationally while writing progresses. The *static* schema plans the entire plot before writing. We summarize the contributions of the paper as follows:

- We propose a *plan-and-write* framework that leverages storylines to improve the diversity and coherence of the generated story. Two strategies: *dynamic* and *static* planning are explored and compared under this framework.
- We develop evaluation metrics to measure the diversity of the generated stories, and conduct novel analysis to examine the importance of different aspects of stories for human evaluation.
- Experiments show that the proposed plan-and-write model generates more diverse, coherent, and on-topic stories than those without planning [2].

Plan-and-Write Storytelling

In this paper, we propose a plan-and-write framework to generate stories from given titles. We posit that storytelling systems can benefit from storyline planning to generate more coherent and on-topic stories. An additional benefit of the plan-and-write schema is that human and computer can interact and collaborate on the (abstract) storyline level, which can enable many potentially enjoyable interactions. We formally define the input, output, and storyline of our approach as follows.

Problem Formulation

Input: A title $\mathbf{t} = \{t_1, t_2, ..., t_n\}$ is given to the system to constrain writing, where t_i is the i-th word in the title.

Output: The system generates a story $\mathbf{s} = \{s_1, s_2, ..., s_m\}$ based on a title, where s_i denotes a sentence in the story.

Storyline: The system plans a storyline $\mathbf{l} = \{l_1, l_2, ..., l_m\}$ as an intermediate step to represent the plot of a story. We use a sequence of words to represent a storyline, therefore, l_i denotes a word in a storyline.

Given a title, the plan-and-write framework always plans a storyline. We explore two variations of this framework: the *dynamic* and the static schema.

Storyline Preparation

To obtain training data for the storyline planner, we extract sequences of words from existing story corpora to compose storylines. Specifically, we extract one word from each sentence of a story to form a storyline[3]. We adopt the RAKE algorithm (Rose et al. 2010), which combines several word frequency based and graph-based metrics to weight the importance of the words. We extract the most important word from each sentence as a story's storyline.

Methods

We adopt neural generation models to implement our plan-and-write framework, as they have been shown effective in many text generation tasks such as machine translation (Bahdanau, Cho, and Bengio 2015), and dialogue systems (Shang, Lu, and Li 2015). Figure 1 demonstrates the workflow of our framework. We now describe the two plan-and-write strategies we explored.

Dynamic Schema

The dynamic schema emphasizes flexibility. As shown in Figure 2a, it generates the next word in the storyline and the next sentence in the story at each step. In both cases, the existing storyline and previously generated sentences are given to the model to move one step forward.

Storyline Planning The storyline is planned out based on the context (the title and previously generated sentences are taken as context) and the previous word in the storyline. We formulate it as a content-introducing generation problem, where the new content (the next word in the storyline) is generated based on the context and some additional information (the most recent word in the storyline). Formally, let $\mathbf{ctx} = [\mathbf{t}, \mathbf{s}_{1:i-1}]$ denotes the context, where $s_{1:i-1}$ denotes for the first i-1 sentences in the story. We model $p(l_i|\mathbf{ctx}, l_{i-1}; \theta)$. We implement the content-introducing method proposed by Yao et al. [2017], which first encodes context into a vector using a bidirectional gated recurrent unit (BiGRU), and then incorporates the auxiliary information, in this case the previous word in the storyline, into the decoding process. Formally, hidden vectors for context are computed as $\widetilde{\mathbf{h}}_{ctx} = Encode_{ctx}(\mathbf{ctx}) = [\overrightarrow{\mathbf{h}_{ctx}}; \overleftarrow{\mathbf{h}_{ctx}}]$, where $\overrightarrow{\mathbf{h}_{ctx}}$ and $\overleftarrow{\mathbf{h}_{ctx}}$ are the hidden vectors produced by a forward and a backward GRU, respectively. $[;]$ denotes element-wise concatenation. The conditional probability is computed as:

$$
\begin{aligned}
&h_y = \mathrm{GRU}(\mathrm{BOS}, C_{att}), h_w = \mathrm{GRU}(l_{i-1}, C_{att}) \\
&h_y^{'} = \tanh(W_1 h_y), h_w^{'} = \tanh(W_2 h_w) \\
&k = \sigma(W_k[h_y^{'}; h_w^{'}]) \\
&p(l_i|\mathbf{ctx}, l_{i-1}) = g(k \circ h_y + (1-k) \circ h_w)
\end{aligned}
$$

BOS denotes the beginning of decoding. C_{att} represents the attention-based context computed from $\widetilde{\mathbf{h}}_{ctx}$. g($\cdot$) denotes a multilayer perceptron (MLP).

Story Generation The story is generated incrementally by planning and writing alternately. We formulate it as another content-introducing generation problem which generates a

[1] Some discussions on Quora: https://www.quora.com/How-many-times-does-a-writer-edit-his-first-draft

[2] Code and appendix will be available at https://bitbucket.org/VioletPeng/language-model

[3] For this pilot study, we assume each word l_i in a storyline corresponds to a sentence s_i in a story.

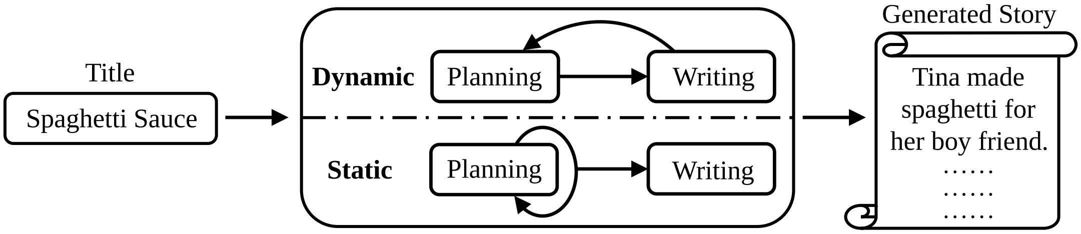
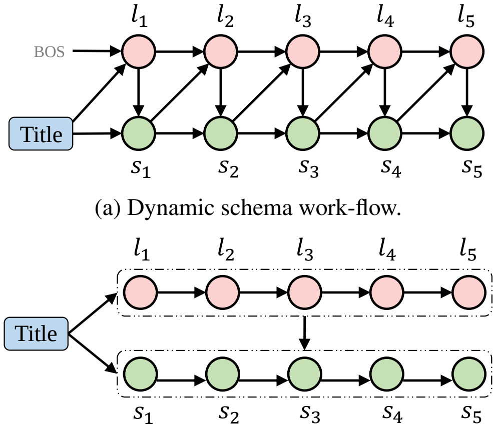


Figure 1: An overview of our system.


l_1 l_2 l_3 l_4 l_5
BOS
Title
s_1 s_2 s_3 s_4 s_5
l_1 l_2 l_3 l_4 l_5
Title
s_1 s_2 s_3 s_4 s_5

(a) Dynamic schema work-flow.

(b) Static schema work-flow.

Figure 2: An illustration of the dynamic and static plan-and-write work-flow. l_i denotes a *word* in a storyline and s_i denotes a *sentence* in a story.

story sentence based on both the context and an additional storyline word as a cue. The model structure is exactly the same as for storyline generation. However, there are two differences between storyline and story generation. On one hand, the former aims to generate a word while the latter generates a variable-length sequence. On the other hand, the auxiliary information they use is different.

Formally, the model is trained to minimize the negative log-probability of training data:

$$\mathcal{L}(\theta)_{dyna} = \frac{1}{N}\sum_{j=1}^{N}\left[-\log\prod_{i=1}^{m} p(s_i|\mathbf{ctx}, l_i)\right]_j \qquad (1)$$

where N is the number of stories in training data; m denotes the number of sentences in a story. Given the extracted storylines as described in the previous Section, the storyline and story generation models are trained separately. End-to-end generation is conducted in a pipeline fashion.

Static Schema

The static schema is inspired by sketches that writers usually draw before they flesh out the whole story. As illustrated in Figure 2b, it first generates a whole storyline which does not change during story writing. This sacrifices some flexibility in writing, but could potentially enhance story coherence as it provides "look ahead" for what happens next.

Storyline Planning Differing from the dynamic schema, storyline planning for static schema is solely based on the title $\mathbf{t}$. We formulate it as a conditional generation problem, where the probability of generating each word in a storyline depends on the previous words in the storyline and the title. Formally, we model $p(l_i|\mathbf{t}, l_{1:i-1}; \theta)$. We adopt a sequence-to-sequence (Seq2Seq), conditional generation model that first encodes the title into a vector using a bidirectional long short-term memory network (BiLSTM), and generates words in the storyline using another single-directional LSTM. Formally, the hidden vector $\tilde{\mathbf{h}}$ for a title is computed as $\tilde{\mathbf{h}} = Encode(\mathbf{t}) = [\overrightarrow{\mathbf{h}}; \overleftarrow{\mathbf{h}}]$, and the conditional probability is given by:

$$p(l_i|\mathbf{t}, l_{1:i-1}; \theta) = g(\text{LSTM}_{\text{att}}(\tilde{\mathbf{h}}, l_{i-1}, \mathbf{h}^{\text{dec}}_{i-1}))$$

where LSTM_{att} denotes a cell of the LSTM with attention mechanism (Bahdanau, Cho, and Bengio 2015); $\mathbf{h}^{\text{dec}}_{i-1}$ stands for the decoding hidden state. g(·) again denotes a MLP.

Story Generation The story is generated after the full storyline is planned. We formulate it as another conditional generation problem. Specifically, we train a Seq2Seq model that encodes both the title and the planned storyline into a low-dimensional vector by first concatenating them with a special symbol <EOT> in between, and encode them with BiLSTMs: $\tilde{\mathbf{h}}_{tl} = Encode_{tl}([\mathbf{t}, \mathbf{l}]) = [\overrightarrow{\mathbf{h}_{tl}}; \overleftarrow{\mathbf{h}_{tl}}]$. The Seq2Seq model is then trained to minimize the negative log-probability of the stories in the training data:

$$\mathcal{L}(\theta)_{static} = \frac{1}{N}\sum_{j=1}^{N}\left[-\log\prod_{i=1}^{m} p(s_i|\tilde{\mathbf{h}}_{tl}, s_{1:i-1})\right]_j \qquad (2)$$

Storyline Optimization

One common problem for neural generation models is the repetition in generated results (Li et al. 2016). We observe repetition initially in both the generated storyline (repeated words) and story (repeated phrases and sentences). An advantage of the storyline layer is that given the compact and interpretable representation of the storyline, we can easily apply heuristics to reduce repetition[4]. Specifically, we forbid any word to appear twice when generating a storyline.

Experimental Setup

Dataset

We conduct the experiments on the ROCStories corpus (Mostafazadeh et al. 2016a). It contains 98,162 short

[4] It is important to avoid repetition in the generated stories too. However, it is hard to automatically detect repetition in stories. Optimizing storylines can indirectly reduce repetition in stories.

Number of Stories	$98,161$
Vocabulary size	$33,215$
Average number of words	50

Table 2: Statistics of the ROCStories dataset.

commonsense stories as training data, and additional 1,817 stories for development and test, respectively. The stories in the corpus are five-sentence stories that capture a rich set of causal and temporal commonsense relations between daily events, making them a good resource for training storytelling models. Table 2 shows the statistics of ROCStories dataset. Since only the training set of the ROCStories corpus contains titles, which we need as input. We split the original training data into 8:1:1 for training, validation, and testing.

Baselines

To evaluate the effectiveness of the plan-and-write framework, we compare our methods against representative baselines without a planning module.

Inc-S2S denotes the incremental sentence-to-sentence generation baseline, which creates stories by generating the first sentence from a given title, then generating the i-th sentence from the title and the previously generated i-1 sentences. This resembles the *dynamic* schema without planning. We use a Seq2Seq model with attention (Bahdanau, Cho, and Bengio 2015) to implement the Inc-S2S baseline, where the sequence to sequence model is trained to generate the next sentence based on the context.

Cond-LM denotes the conditional language model baseline, which straightforwardly generates the whole story word by word from a given title. Again we use a Seq2Seq model with attention as our implementation of the conditional language model, where the sequence to sequence model is trained to generate the whole story based on the title. It resembles our *static* schema without planning.

Hyper-parameters

As all of our baselines and the proposed methods are RNN-based conditional generation models, we conduct the same set of hyper-parameter optimization for them. We train all the models using stochastic gradient descent (SGD). For the encoder and decoder in our generation models, we tune the hyper-parameters of the embedding and hidden vector dimensions and the dropout rate by grid search. We randomly initialize the word embeddings and tune the dimensions in the range of [100, 200, 300, 500] for storyline generation and [300, 500, 1000] for story generation. We tune the hidden vector dimensions in the range of [300, 500, 1000]. The embedding and hidden vector dropout rates are all tuned from 0 to 0.5, step by 0.1. We tune all baselines and proposed models based on BLEU scores (Papineni et al. 2002) on the validation set. Details of the best hyper-parameter values for each setting are given in Appendix.

Evaluation Metrics

Objective metrics. Our goal is generating human-like stories, which can pass the Turing test. Therefore, the evaluation metrics based on n-gram overlap such as BLEU are not suitable for our task[5]. To better gauge the quality of our methods, we design novel automatic evaluation metrics to evaluate the generation results at scale. Since neural generation models are known to suffer from generating repetitive content, our automatic evaluation metrics are designed to quantify diversity across the generated stories. We design two measurements to gauge inter- and intra-story repetition. For each sentence position i, the inter-story r_e^i and intra-story r_a^i repetition rate are computed as follows:

$$\begin{aligned} r_e^i &= 1 - \frac{T(\sum_{j=1}^{N} s^{ji})}{T_{all}(\sum_{j=1}^{N} s^{ji})} \\ r_a^i &= \frac{1}{N}\sum_{j=1}^{N}\left[\frac{\sum_{k=1}^{i-1} T(s^i \cap s^k)}{(i-1) * T(s^i)}\right]^j \end{aligned} \tag{3}$$

where $T(\cdot)$ and $T_{all}(\cdot)$ denote the number of distinct and total trigrams[6], respectively. s^{ji} stands for the i-th sentence in j-th story; $s^i \cap s^k$ is the distinct trigram intersection set between sentence s^i and s^k. Naturally, r_e^i demonstrates the repetition rate between stories at sentence position i; r_a^i embodies the average repetition of sentence s^i comparing with former sentences in a story.

We compute the aggregate scores as follows:

$$\begin{aligned} r_e^{agg} &= 1 - \frac{T(\sum_{j=1}^{N}\sum_{i=1}^{m} s^{ji})}{T_{all}(\sum_{j=1}^{N}\sum_{i=1}^{m} s^{ji})} \\ r_a^{agg} &= \frac{1}{m}\sum_{i=1}^{m} r_a^i \end{aligned} \tag{4}$$

where $\sum_{j=1}^{N}\sum_{i=1}^{m} s^{ji}$ is the set of N stories with m sentences. In our experiments, we set $m = 5$. r_e^{agg} indicates the overall repetition of all stories.

Subjective metrics. For a creative generation task such as story generation, reliable automatic evaluation metrics to assess aspects such as interestingness, coherence are lacking. Therefore, we rely on human evaluation to assess the quality of generation. We conduct pairwise comparisons, and provide users two generated stories, asking them to choose the better one. We consider four aspects: *fidelity* (whether the story is on-topic with the given title), *coherence* (whether the story is logically consistent and coherent), *interestingness* (whether the story is interesting) and *overall user preference* (how do users like the story). All surveys were collected on Amazon Mechanical Turk (AMT).

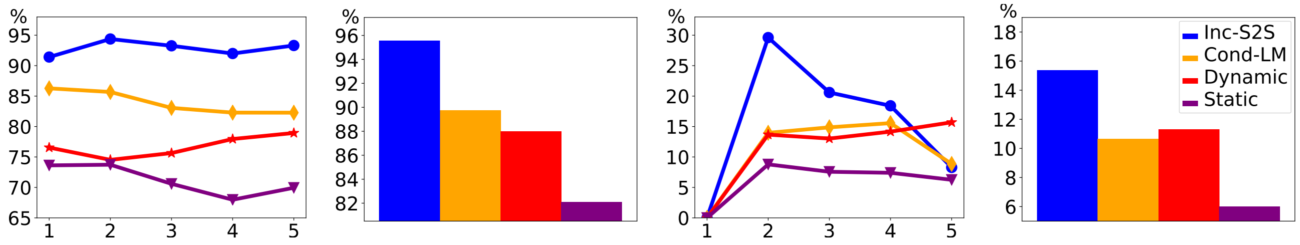
Results and Discussion

Objective evaluation

We generate 9816 stories based on the titles in the held-out test set, and compute the repetition ratio (the lower, the better) as described in Eq. 3 and Eq. 4 to evaluate

[5] Our plan-and-write methods also improve BLEU scores over the baseline methods, more details can be found in Appendix.

[6] We also conduct the same computation for four and five-grams and observed the same trends. The Spearman correlation between this measurement and human rating is 0.28.



(a) Inter-story repetition curve by sentence.

(b) Inter-story aggregate repetition scores.

(c) Intra-story repetition curve by sentence.

(d) Intra-story aggregate repetition scores.

Figure 3: Inter- and intra-story repetition rates by sentences (curves) and for the whole stories (bars), the lower the better. As reference points, the aggregate repetition rates on the human-written training data are 34% and 0.3% for the inter- and intra-story measurements respectively.

Choice %	Dynamic *vs* Inc-S2S			Static *vs* Cond-LM			Dynamic *vs* Static		
	Dyna.	Inc.	Kappa	Static	Cond.	Kappa	Dyna.	Static	Kappa
Fidelity	**35.8**	12.9	0.42	**38.5**	16.3	0.42	21.47	**38.00**	0.30
Coherence	**37.2**	28.6	0.30	**39.4**	32.3	0.35	28.27	**49.47**	0.36
Interestingness	**43.5**	26.7	0.31	**39.5**	35.7	0.42	34.40	**42.60**	0.35
Overall Popularity	**42.9**	27.0	0.34	**40.9**	34.2	0.38	30.07	**50.07**	0.38

Table 3: Human evaluation results on four aspects: fidelity, coherence, interestingness, and overall user preference. Dyna., Inc., and Cond. is the abbreviation for Dynamic schema, Inc-S2S, and Cond-LM respectively. We also calculate the Kappa coefficient to show the inter-annotator agreement.

the diversity of the generated system. As is shown in Figure 3, the proposed plan-and-write framework significantly reduces the repetition rate and generates more diverse stories. For inter-story repetition, plan-and-write methods significantly outperform all non-planning methods on individual sentences and aggregate scores. For the intra-story repetition rate, plan-and-write methods outperform their corresponding non-planning baselines on aggregate scores. However, the dynamic schema generates more repetitive final sentences than the baselines.

Subjective evaluation

For human evaluation, we randomly sample 300 titles from the test data, and present a story title and two generated stories at a time[7] to the evaluators and ask them to decide which of the two stories is better[8]. There are 233 Turkers[9] participated in the evaluation. Specifically, 69, 77, and 87 Turkers evaluate the comparison between Dynamic and Inc-S2S, Static and Cond-LM, Dynamic and Static, respectively.

Table 3 demonstrates the results of human evaluation. Similar to the automatic evaluation results, both dynamic and static schema significantly outperform their counterpart baseline in all evaluation aspects, thus demonstrating the effectiveness of the proposed plan-and-write framework. Among them, the static schema shows the best results.

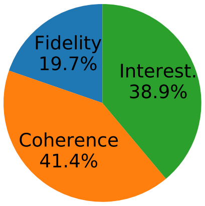


Figure 4: The regression coefficient that shows which aspect is more important in human evaluation of stories.

To understand why people prefer one story over another, we analyze how people weigh the three aspects (fidelity, coherence, and interestingness) in their preference for stories. We train a linear regression using the three aspects' scores as features to predict the overall score. We fit the regression with all human assessments we collected. The weight assigned to each aspect reflects their relative importance. As evident in Figure 4, coherence and interestingness play important roles in the human evaluation, and fidelity is less important.

Analysis

The previous sections examine the overall performance of our methods quantitatively. In this section, we qualitatively analyze our methods with a focus on comparing the dynamic and static schema.

Storyline analysis. First, we measure the quality of the generated storylines, and the correlations between a storyline and the generated story. We use BLEU scores to measure the quality of the storylines, and an embedding-based metric (Liu et al. 2016a) to estimate the average greedy matching score l-s between storyline words and generated

[7] We compare the plan-and-write methods with their corresponding baselines and with each other. For fairness, the two stories are pooled and randomly permuted. Five judgments are required to reduce the variance in estimation.

[8] The four aspects are each evaluated.

[9] We applied qualification filters that only allow users who have at least 500 previous jobs and had greater than 98% acceptance rate to participate in our survey.

Title: Computer		
Baselines	Inc-S2S	Tom's computer broke down. He needed to buy a new computer. He decided to buy a new computer. Tom bought a new computer. Tom was able to buy a new computer.
	Cond-LM	The man bought a new computer. He went to the store. He bought a new computer. He bought the computer. He installed the computer.
Dynamic	Storyline	needed → money → computer → bought → happy
	Story	John needed a computer for his birthday. He worked hard to earn money. John was able to buy his computer. He went to the store and bought a computer. John was happy with his new computer.
Static	Storyline	computer → slow → work → day → buy
	Story	I have an old computer. It was very slow. I tried to work on it but it wouldn't work. One day, I decided to buy a new one. I bought a new computer .
Title: The Virus		
Baselines	Inc-S2S	His computer was fixed and he fixed it. John got a new computer on his computer. John was able to fix it himself. John was able to fix his computer and was able to fix his computer. John was able to fix his computer and had a virus and was able to fix his computer.
	Cond-LM	Tim was working on a project. He was working on a project. Tim was working on a project. The project was really good. Tim was able to finish the project.
Dynamic	Storyline	computer → use → anywhere → house → found
	Story	I was working on my computer today. I was trying to use the computer. I couldn't find it anywhere. I looked all over the house for it. Finally, I found it.
Static	Storyline	work → fix → called → found → day
	Story	I had a virus on my computer. I tried to fix it but it wouldn't work. I called the repair company. They came and found the virus. The next day, my computer was fixed.

Table 4: Case studies of generated storylines and stories.

Title / Problem	**Story**
Taxi / off-topic	I got a new car. It was one day. I decided to drive to the airport. I was driving for a long time. I had a great time .
Cut / repetitive	Anna was cutting her nails. She cut her finger and cut her finger. Then she cut her finger. It was bleeding! Anna had to bandage her finger.
Eight glasses/ inconsistent	Joe needed glasses. He went to the store to buy some. He did n't have any money. He found a pair that he liked. He bought them.

Table 5: Example stories that demonstrate the typical problems of the current systems.

Method	l-B1	l-B2	l-s
Dynamic	6.46	0.79	0.88
Static	9.53	1.59	0.89

Table 6: The storyline BLEU score (only BLEU-1 and BLEU-2) and the correlation of storyline-story **l-s**.

story sentences. Concretely, a storyline word is greedily matched with each token in a story sentence based on the cosine similarity of their word embeddings[10]. The highest cosine score is regarded as the correlation between them.

Table 6 shows the results. We can see that the static schema generates storylines with higher BLEU scores. It also generates stories that have a higher correlation with the storylines (higher **l-s** score[11]). This indicates that with better storylines (higher BLEU score), it is easier to generate more relevant and coherent stories. This partially explains why the static schema performs better than the dynamic schema.

Case study. We further present two examples in Table 4 to intuitively compare the plan-and-write methods and the baselines[12]. In both examples, the baselines without planning components tend to generate repetitive sentences that do not exhibit much of a story progression. In contrast, the plan-and-write methods can generate storylines that follow a reasonable flow, and thus help generate coherent stories with less repetition. This demonstrates the ability of the plan-and-write methods. In the second example, the storyline generated by the dynamic schema is not very coherent and thus significantly affects story quality. This reflects the importance of storyline planning in our framework.

Error analysis. To better understand the limitation of our best system, we manually reviewed 50 titles and the corresponding generated stories from our static schema to conduct error analysis. The three major problems are: off-topic,

[10] For fairness, We adopt the pre-trained Glove embedding to measure the correlation. `http://nlp.stanford.edu/data/glove.840B.300d.zip`

[11] There are 75% and 78% storyline words appear in the generated stories in the dynamic and static schema, respectively.

[12] More examples please see our live demo at `http://cwc-story.isi.edu/`

repetitive, and logically inconsistent. We show three examples, one for each category, in Table 5 to illustrate the problems. We can see that the current system is already capable of generating grammatical sentences that are coherent within a local context. However, generating a sequence of coherent and logically consistent sentences is still an open challenge.

Related work

Story Planning

Automatic story generation efforts date back to the 1970s (Meehan 1977). Early attempts focused on composing a sensible plot for a story. Symbolic planning systems (Porteous and Cavazza 2009; Riedl and Young 2010) attempted to select and sequence character actions according to specific success criteria. Case-based reasoning systems (Turner 1994; Gervas et al. 2005; Montfort 2006) adapted prior story plots (cases) to new storytelling requirements. These traditional approaches were able to produce impressive results based on hand-crafted, well-defined domain models, which kept track of legal characters, their actions, narratives, and user interest. However, the generated stories were restricted to limited domains.

To tackle the problem of restricted domains, some work attempted to automatically learn domain models. Swanson and Gordon [2012] mined millions of personal stories from the Web and identified relevant existing stories in the corpus. Li et al. [2013] used a crowd-sourced corpus of stories to learn a domain model that helped generate stories in unknown domains. These efforts stayed at the level of story plot planning without surface realization.

Event Structures for Storytelling

There is a line of research focusing on representing story event structures (Mostafazadeh et al. 2016b; McDowell et al. 2017). Rishes et al. [2013] presents a model that reproduce different versions of a story from its symbolic representation. Pichotta and Mooney [2016] parse a large collection of natural language documents, extract sequences of events, and learn statistical models of them. Some recent work explored story generation with additional information (Bowden et al. 2016; Peng et al. 2018; Guan, Wang, and Huang 2019). Visual storytelling (Huang et al. 2016; Liu et al. 2016b; Wang et al. 2018) aims to generate human-level narrative language from a sequence of images. Jain et al. [2017] addresses the task of coherent story generation from independent textual descriptions. Unlike this line of work, we learn to *automatically generate* storylines to help generate coherent stories.

Martin et al.; Xu et al. [2018; 2018] are the closest work to ours, which decomposed story generation into two steps: story structure modeling and structure-to-surface generation. However, Martin et al. [2018] did not conduct experiments on full story generation. Xu et al. [2018] is a concurrent work which is similar to our dynamic schema. Their setting assumes story prompts as inputs, which is more specific than our work (which only requires a title). Moreover, we explore two planning strategies: dynamic schema and static schema, and show the latter works better.

Neural Story Generation

Recently, deep learning models have been demonstrated effective in natural language generation tasks (Bahdanau, Cho, and Bengio 2015; Merity, Keskar, and Socher 2018) In story generation, prior work has proposed to use deep neural networks to capture story structures and generate stories. Khalifa, Barros, and Togelius [2017] argue that stories are better generated using recurrent neural networks (RNNs) trained on highly specialized textual corpora, such as a body of work from a single, prolific author. Roem et al. [2017] use skip-thought vectors (Kiros et al. 2015) to encode sentences and model relations between the sentences. Jain et al. [2017] explore generating coherent stories from independent textual descriptions based on two conditional text-generation methods: statistical machine translation and sequence-to-sequence models. Fan, Lewis, and Dauphin [2018] proposes a hierarchical generation strategy to generate stories from prompts to improve coherence. However, we consider storylines are different from prompts as they are not naturally language sentences. They are some structured outline of stories. We employ neural network-based generation models for our plan-and-write generation. The focus, however, is to introduce storyline planning to improve the quality of generated stories, and compare the effect of different storyline planning strategies on story generation.

Conclusion and Future Work

In this paper, we propose a *plan-and-write* framework that generates stories from given titles with explicit storyline planning. We explore and compare two plan-and-write strategies: dynamic schema and static schema, and show that they both outperform the baselines without planning components. The static schema performs better than the dynamic schema because it plans the storyline holistically, thus tends to generate more coherent and relevant stories.

The current plan-and-write models use a sequence of words to approximate a storyline, which simplifies many meaningful structures in a real story plot. We plan to extend the exploration to richer representations, such as entity, event, and relation structures, to depict story plots. We also plan to extend the plan-and-write framework to generate longer documents. The current framework relies on storylines automatically extracted from story corpora to train the planning module. In the future, we will explore the storyline induction and joint storyline and story generation to avoid error propagation in the current pipeline generation system.

Acknowledgements

We thank the anonymous reviewers for the useful comments. This work is supported by Contract W911NF-15-1-0543 with the US Defense Advanced Research Projects Agency (DARPA), the National Key Research and Development Program of China (No. 2017YFC0804001), and the National Science Foundation of China (Nos. 61876196 and 61672058).

References

[2010] Alarcon, D. 2010. *The Secret Miracle: The Novelist's Handbook*. St. Martin's Griffin.

[2015] Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by learning to align and translate. In *ICLR*.

[2016] Bowden, K. K.; Lin, G. I.; Reed, L. I.; Tree, J. E. F.; and Walker, M. A. 2016. M2d: Monolog to dialog generation for conversational story telling. In *ICIDS*.

[2018] Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical neural story generation. In *ACL*.

[2005] Gervas, P.; Diaz-Agudo, B.; Peinado, F.; and Hervas, R. 2005. Story plot generation based on CBR. In *KBS*.

[2019] Guan, J.; Wang, Y.; and Huang, M. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *AAAI*.

[2016] Huang, T.-H. K.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Devlin, J.; Agrawal, A.; Girshick, R.; He, X.; Kohli, P.; Batra, D.; et al. 2016. Visual storytelling. In *NAACL*.

[2017] Jain, P.; Agrawal, P.; Mishra, A.; Sukhwani, M.; Laha, A.; and Sankaranarayanan, K. 2017. Story generation from sequence of independent short descriptions. In *KDD WS*.

[2017] Khalifa, A.; Barros, G. A.; and Togelius, J. 2017. Deeptingle. In *arXiv preprint arXiv:1705.03557*.

[2015] Kiros, R.; Zhu, Y.; Salak, R.; Zemel, R.; Urtasun, R.; Torral, A.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*.

[1987] Lebowitz, M. 1987. Planning stories. In *CogSci*.

[2013] Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story generation with crowdsourced plot graphs. In *AAAI*.

[2016] Li, J.; Monroe, W.; Ritter, A.; Jurafsky, D.; Galley, M.; and Gao, J. 2016. Deep reinforcement learning for dialogue generation. In *EMNLP*.

[2016a] Liu, C.-W.; Lowe, R.; Serban, I. V.; Noseworthy, M.; Charlin, L.; and Pineau, J. 2016a. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *EMNLP*.

[2016b] Liu, Y.; Fu, J.; Mei, T.; and Chen, C. W. 2016b. Storytelling of photo stream with bidirectional multi-thread recurrent neural network. In *arXiv preprint arXiv:1606.00625*.

[2018] Martin, L.; Ammanabrolu, P.; Hancock, W.; Singh, S.; Harrison, B.; and Riedl, M. 2018. Event representations for automated story generation with deep neural nets. In *AAAI*.

[2017] McDowell, B.; Chambers, N.; Ororbia II, A.; and Reitter, D. 2017. Event ordering with a generalized model for sieve prediction ranking. In *IJCNLP*.

[1977] Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. In *IJCAI*.

[2018] Merity, S.; Keskar, N. S.; and Socher, R. 2018. Regularizing and optimizing lstm language models. In *ICLR*.

[2006] Montfort, N. 2006. Natural language generation and narrative variation in interactive fiction. In *AAAI WS*.

[2016a] Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016a. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL*.

[2016b] Mostafazadeh, N.; Grealish, A.; Chambers, N.; Allen, J.; and Vanderwende, L. 2016b. CaTeRS: Causal and temporal relation scheme for semantic annotation of event structures. In *NAACL Workshop*.

[2016] Mou, L.; Song, Y.; Yan, R.; Li, G.; Zhang, L.; and Jin, Z. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *COLING*.

[2017] Nayak, N.; Hakkani-Tur, D.; Walker, M.; and Heck, L. 2017. To plan or not to plan? discourse planning in slot-value informed sequence to sequence models for language generation. In *Interspeech*.

[2002] Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

[2018] Peng, N.; Ghazvininejad, M.; May, J.; and Knight, K. 2018. Towards controllable story generation. In *NAACL Workshop*.

[2001] Perez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. In *JETAI*.

[2016] Pichotta, K., and Mooney, R. J. 2016. Learning statistical scripts with LSTM recurrent neural networks. In *AAAI*.

[2009] Porteous, J., and Cavazza, M. 2009. Controlling narrative generation with planning trajectories: The role of constraints. In *ICIDS*.

[2010] Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. In *JAIR*.

[2013] Rishes, E.; Lukin, S. M.; Elson, D. K.; and Walker, M. A. 2013. Generating different story tellings from semantic representations of narrative. In *ICIDS*.

[2017] Roem, M.; Koba, S.; Inoue, N.; and Gordon, A. M. 2017. An RNN-based classifier for the story cloze test. In *LSDSem*.

[2010] Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. Automatic keyword extraction from individual documents. In *Text Mining: Applications and Theory*.

[2015] Shang, L.; Lu, Z.; and Li, H. 2015. Neural responding machine for short-text conversation. In *ACL*.

[2012] Swanson, R., and Gordon, A. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. In *ACM TiiS*.

[1994] Turner, S. R. 1994. *The creative process: A computer model of storytelling and creativity*. Psychology Press.

[2016] Wang, Z.; He, W.; Wu, H.; Wu, H.; Li, W.; Wang, H.; and Chen, E. 2016. Chinese poetry generation with planning based neural network. In *COLING*.

[2018] Wang, X.; Chen, W.; Wang, Y.-F.; and Wang, W. Y. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling. In *ACL*.

[2018] Xu, J.; Zhang, Y.; Zeng, Q.; Ren, X.; Cai, X.; and Sun, X. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *EMNLP*.

[2017] Yao, L.; Zhang, Y.; Feng, Y.; Zhao, D.; and Yan, R. 2017. Towards implicit content-introducing for generative short-text conversation systems. In *EMNLP*.

Appendix

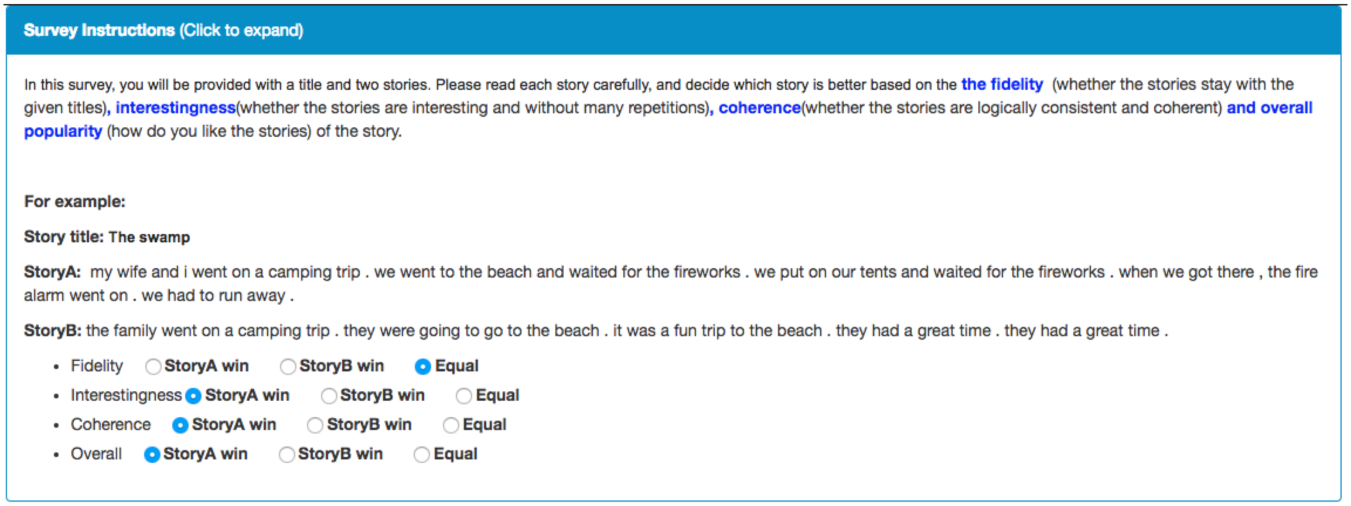
In this section, we describe the best hyper-parameter values for each setting in Table 7 and present BLEU score in Table 8. To provide an intuitive understanding of human evaluation process, we show a snapshot of the survey on AMT in Figure 5.

Method	E-Dim	H-Dim	E-Drop	H-Drop
Inc-S2S	500	500	0.0	0.5
Cond-LM	500	1000	0.4	0.4
Dynamic-l	500	500	0.0	0.5
Dynamic-s	500	500	0.0	0.5
Static-l	500	1000	0.4	0.1
Static-s	500	1000	0.2	0.1

Table 7: Best hyper-parameter settings for the baselines and our models. "-Dim" denotes dimension, "-Drop" represents dropout rate, and "E" and "H" denote the embedding and hidden layer. "-l" and "-s" stand for storyline and story respectively.

Method	B-1	B-2	B-3	B-4
Inc-S2S	25.13	10.22	4.65	2.35
Cond-LM	28.07	11.62	5.11	2.55
Dynamic-W.O.	26.16	10.39	4.62	2.32
Static-W.O.	27.16	12.21	6.04	3.27
Dynamic	28.47	11.49	5.21	2.62
Static	28.20	12.80	6.36	3.44

Table 8: BLEU scores in testing set. Suffix "-W.O." denotes NO optimization for storylines.



Survey Instructions (Click to expand)

In this survey, you will be provided with a title and two stories. Please read each story carefully, and decide which story is better based on the **the fidelity** (whether the stories stay with the given titles)**, interestingness**(whether the stories are interesting and without many repetitions)**, coherence**(whether the stories are logically consistent and coherent) **and overall popularity** (how do you like the stories) of the story.

For example:

Story title: The swamp

StoryA: my wife and i went on a camping trip . we went to the beach and waited for the fireworks . we put on our tents and waited for the fireworks . when we got there , the fire alarm went on . we had to run away .

StoryB: the family went on a camping trip . they were going to go to the beach . it was a fun trip to the beach . they had a great time . they had a great time .

- Fidelity ○ **StoryA win** ○ **StoryB win** ● **Equal**
- Interestingness ● **StoryA win** ○ **StoryB win** ○ **Equal**
- Coherence ● **StoryA win** ○ **StoryB win** ○ **Equal**
- Overall ● **StoryA win** ○ **StoryB win** ○ **Equal**

Figure 5: A snapshot of the survey on AMT for human evaluation.

.