Story Generation with Rich Details

Fangzhou Zhai[†], **Vera Demberg**[†,‡], and **Alexander Koller**[†]
[†] Dept. of Language Science and Technology
[‡] Dept. of Computer Science
Saarland Informatics Campus
Saarland University
{fzhai, vera, koller}@coli.uni-saarland.de

Abstract

Automatically generated stories should be not only coherent, but also interesting. Thus apart from realizing a story line, the text also have to include rich details to engage the readers. We propose a model that features two different generation components: (1) an ***outliner***, which proceeds the main story line to establish global coherence; and (2) a ***detailer***, which supplies relevant details to the story in a locally coherent manner. Human evaluation show that our model substantially improves the informativeness of generated text while retaining its coherence, outperforming a number of baselines.

1 Introduction

Story generation is the task of automatically crafting stories. Recent neural story generation systems have been able to produce coherent stories. Global coherence could be established by conditioning language models on the longer-term intention of the text. One could provide a topic (see, e.g. Fan et al. (2018), Fan et al. (2019)), a list of events (see, e.g. Zhai et al. (2019), Martin et al. (2018), Ammanabrolu et al. (2019)) or entities (see, e.g. Kiddon et al. (2016), Clark et al. (2018)) etc., to guide the generation process. But apart from being globally coherent, a story also needs to be interesting to engage its readers.

yesterday i went grocery shopping . i made a list of my list and drove to the grocery store . when i entered the store , i grabbed a shopping cart and pushed the cart down to the meat aisle . i got all my items , and crossed items on my list . i went to the checkout register and paid for my groceries . i put my groceries in my cart and left .

Figure 1: A story about *grocery shopping* generated by Zhai et al. (2019), which is globally coherent but really boring.

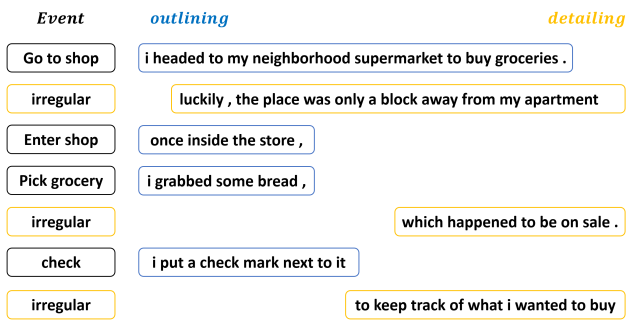
Event
outlining
detailing
Go to shop
i headed to my neighborhood supermarket to buy groceries .
irregular
luckily , the place was only a block away from my apartment
Enter shop
once inside the store ,
Pick grocery
i grabbed some bread ,
irregular
which happened to be on sale .
check
i put a check mark next to it
irregular
to keep track of what i wanted to buy

Figure 2: Sample Data from INSCRIPT. The left column list the event annotations, and on the right are the text ***segments*** corresponding to each event annotation. The story naturally decomposes to be an alternating effort between ***outlining*** and ***detailing***.

Figure 1 shows a story about *grocery shopping*. The story is globally coherent as a narration of grocery shopping, but no one would be interested by such a story: it hardly provides any further information than what the topic *grocery shopping* already indicates.

Our key observation is that, story writing could be seen as the joint efforts of two different components, and a story generation system would benefit from modelling them differently. (1) ***Outlining***, where a story realizes a (usually) linear story line step by step, thus establishing the global coherence of the text. (2) ***Detailing***, where the author supplies details about some of the steps in the story line such that the story becomes informative and interests its reader. A neural story generation system that conditions the generation on the story line accomplishes the former but may well fail the latter.

We propose a model that generates stories with rich details, by addressing these two components differently: an *outliner* realizes consecutive events to establish the story line, whereas a *detailer* supplies additional details at specific points along the story line. Human evaluations show that our system significantly outperforms various baselines in terms of informativeness, its outputs maintaining their coherence as stories about daily activities.

2 Model

2.1 Task and Data

We work with InScript (Modi et al. (2016)), which contains around 100 stories for each of 10 daily activities like *grocery shopping*. Most of its verbal phrases are annotated with event types. There are two categories of events. **(1)** *Regular* events are closely related to the specific daily activity . These events correspond to the fundamental steps of the activity, like '*go to the shop, pick groceries, pay*'. **(2)** *Irregular* events are not directly related to the core content of that activity, but instead supply relevant details (see figure 2). Regular events correspond to the outlining component, whereas irregular events correspond to the detailing component.

On average, there are 20 regular event types per activity. Events with labels '*unrelated* (not related to the main activity), *non-script* (related but not a part of the activity per se), *unknown*' and *other* are considered *irregular*. Due to the varying nature of irregular events' content, they receive generic annotations. In total, the corpus contains 234k tokens, and realizes 15k event instances. About 40% of these events are irregular. We automatically dissect the stories to assign each event annotation to its corresponding text ***segment*** based on POS tags.

Our system receives a plausible sequence of events as input, which we term an ***agenda***. Agendas are automatically generated by a bi-gram language model trained on the event sequences from the corpus; an agenda includes irregular event items, which specifies locations where additional details are needed. The model generates a story which realizes the agenda.

2.2 Model Specifics

Our neural model is an enriched ATT-SEQ2SEQ model with two different decoders: (1) an ***outliner***, which generates regular segments to instantiate the regular event items in the agenda; and (2) a ***detailer***, which generates irregular segments to supply details to the story. The model generates a story segment by segment, alternating between the decoders to address one event at a time. Thus technically, the neural part of the model receives (1) the input sequence, i.e. segments $s = \langle s_1 \dots s_{i-1} \rangle$ that are already generated and (2) the agenda $a = \langle e_1 \dots e_n \rangle$ that consists of n events in total as input; as the output, it generates the next text segment s_i which corresponds to the current event e_i.

Encoder We encode the input sequence in an agenda-aware manner:

$$\boldsymbol{f_i} = Enc(\varphi_w(s_{<i}); \varphi_e(e_{<i})) \tag{1}$$

Here $\varphi_w(\cdot)$ denotes the word embeddings, which we initialize with pre-trained Glove embeddings (Pennington et al. (2014)); $\varphi_e(\cdot)$ is the event embeddings, which we initialize randomly. The embedding of each token in the history $s_{<i}$ is concatenated (;) with the embedding $\varphi_e(e_i)$ of its corresponding event e_i so the encoding process is aware of the agenda. $Enc(\cdot)$ is a single layer Bi-LSTM sequence encoder. The outcome $\boldsymbol{f_i}$ is a list of vectors, each of which collects the features of its corresponding input token.

Outliner The *outliner* generates the text segment corresponding to the next event e_i in the agenda, if it is regular. Here we use a Bi-LSTM sequence decoder which has access to a dot-product attention (Luong et al. (2015)) over the encoded sequence $\boldsymbol{f_i}$. At each decoding step t it yields a distribution from which one could sample the next token:

$$p_t = Dec(\varphi_w(tok_{t-1}); att(d_{t-1}, \boldsymbol{f_i}); \varphi_e(e_i)) \tag{2}$$

here, d_{t-1} is the inner state of the decoder before generating the t-th token of the current segment; $att(\cdot, \cdot)$ denotes dot-product attention, so $att(d_{t-1}, \boldsymbol{f_i})$ collects information from the encoder; tok_{t-1} is the token generated in the previous step $t-1$; $\varphi_e(e_i)$ denotes the embedding of the target event. The complete segment will be generated with beam-search.

Detailer If the next event e_i is irregular, the *detailer* generates its text segment. Whereas the *outliner* gets a regular event e_i like *pick up groceries* which is informative of the content of the next segment, the *detailer* only gets an *irregular* event type, which incorporates little information. This is the main technical challenge of the paper, which we tackle with the following efforts.
(1) To select the content of the current segment, we condition the decoding process on its most important context: the previous regular event e_- and the successive regular event e_+. Thus at each decoding step t, it produces:

$$q_t = Dec(\varphi_w(tok_{t-1}); att(d_{t-1}, \boldsymbol{f_i}); \varphi_e(e_-); \varphi_e(e_+)) \tag{3}$$

Here, $Dec(\cdot)$ is once again a single layer Bi-LSTM.
(2) The *detailer* also adopts the maximum mutual information (MMI) objective. First used in conjunction with SEQ2SEQ models by Li et al. (2016), the technique promotes the generation of specific, meaningful texts by moderately suppressing generic language generations. The idea is, instead of maximizing data likelihood, one could maximize the mutual information $I(c, s)$ between the context c and the generation s to promote the correspondence between the two, thus improving the informativeness of the text. The MMI decoding objective generalizes to:

$$s = \arg\max_s [log(p(s|c)) - \lambda \cdot log(p(s))] \tag{4}$$

which is the maximum likelihood objective minus a language model term, so it is also termed an *anti-LM* objective. In practice, equivalently, we follow the approach proposed by Li et al. (2016): we keep the maximum likelihood training intact, whereas in the inference phase, we use a pre-trained language model on INSCRIPT to estimate the anti-LM term, and add it to the scoring within beam search. The coefficient λ is set at 0.1.

Overall, our maximization objective is

$$L(d) = \sum_{tok_t:regular} \log p_t(tok_t) + \sum_{tok_t:irregular} \log q_t(tok_t) \tag{5}$$

2.3 Generation

In the inference phase, we perform beam search to generate each segment. The beam size is set to 5 for the *outliner*. For the *detailer*, the scoring also exploits the *anti-LM* term, so a larger beam size is needed to effectively exploit the MMI objective, thus we set its beam size to 100.

3 Experiments

3.1 Set-up, Implementation and Optimization

5% of the stories in INSCRIPT are randomly selected as the validation set. As we use human evaluation, no test set is necessary. The model is implemented with AllenNLP 0.9.0 (Gardner et al. (2017)). Hyper-parameters were chosen by random hyper parameter search (Bergstra and Bengio (2012)). The model is optimized with Adam (Kingma and Ba (2014)) at a learning rate of 7.5×10^{-4}. Dimensions of the encoder and the decoder are both 256. The word embedding size is fixed at 300 as we initialize it with the 840B version of the pre-trained GLOVE. A dropout (Srivastava et al. (2014)) at rate 0.69 is applied to all dense connections. Gradient norms are clipped at 2.0. We use early stopping with a patience of 35 to further regularize the training. Each training session takes on average 4 hours on a Tesla v100, whereas the generation speed is on average 8 tokens/second [1].

3.2 Evaluation

Baselines We evaluate our model against three baselines: **(1)** the GRU-based model from Zhai et al. (2019), as it also takes an ordered agenda as input; **(2)** an ablated version of our model, SINGLE DECODER, which uses only one decoder, the *outliner*, to neutralize the effort to treat regular and irregular contents differently; **(3)** the human authored stories from INSCRIPT, to upper bound the performance. For some intuition, table 1 provides a few sample texts. Other external story generation systems we considered are not suitable either because they do not admit a compatible input format (i.e. taking a linear agenda as input) or that they cannot fully exploit the temporal order information encoded in an linear agenda (such systems tend to perform poorly on INSCRIPT based story generation (see Zhai et al. (2019)).

ZHAI ET AL. (2019)
yesterday i went grocery shopping . i made a list of my list and drove to the grocery store . when i entered the store , i grabbed a shopping cart and pushed the cart down to the meat aisle . i got all my items , and crossed items on my list . i went to the checkout register and paid for my groceries . i put my groceries in my cart and left .
Agenda: *evoking→make list→go to store→enter→take cart→move along sections→take grocery→check list→go to checkout→pay→pack grocery→leave→ story ends*
SINGLE DECODER
yesterday , i went grocery shopping . *i was very excited so i went grocery shopping* . i went to the store *and parked my car* . i grabbed a shopping cart so i went to the produce section . i went to the store so i picked up my list . i got off my list *when i was done* , i went to the cashier and paid for my debit card . i put my bags into my cart and paid for my groceries . then i paid , *i thanked the cashier* and headed to the car . i put the cart in the cart and drove home .
Agenda: *evoking →irregular go to grocery store→irregular→take a shopping cart→move along the sections→get groceries→check bought items off the shopping list→irregular→go to checkout→pay→irregular→pack groceries→get receipt→bring to vehicle→return the shopping cart→leave*
FULL
i went grocery shopping today . i made a list of things that i needed before i left my house . i looked at the grocery store immediately , *i drove to the grocery store that is always empty* . i parked the car , walked inside and grabbed a shopping cart . i proceeded down the produce aisle , *which happened to be on sale* . i went to the back and chose my favorite peanut butter . i went to the cashier , which was a short line . i had to wait in line since there were some people ahead of me . i put my items on the conveyor belt and then she scanned my items . i said hi to the cashier , i paid my bill , and another man placed all my items in my shopping bags . i made sure to put the cart away , too . *once i had everything i needed ,* i then picked up my bags and left the store . *on the way out i decided to rent a redbox movie for the night since the kiosk was right by* .
Agenda: *evoking→make a shopping list →go to grocery store→irregular →take a shopping cart →move along the sections→irregular →get groceries→ go checkout→wait→cashier scan/weigh items→put stuff on the conveyor→irregular→pay→get receipt →wait →bring to vehicle→leave→wait*
HUMAN AUTHOR
yesterday i went grocery shopping . i took my grocery list with me , along with some reusable shopping bags . my grocery list has all the items i want to buy on it . i selected a shopping cart from in front of the store , and went inside . *i put my reusable bags in the cart* . i looked at my list and started in the produce section . i put different vegetables and fruits into my cart . next i wheeled my cart to the cereal aisle and took a box of cereal . i went through the store aisle by aisle and selected my groceries . *each aisle is organized by types of food and non-food items . one aisle has dried pasta , canned tomatoes , rice , and sauce* . i selected a few boxes of pasta and some rice . another aisle carries plastic wrap , trash bags , and aluminum foil . as i went through the store , i kept looking at my list to see what i needed next . when i added each item to my cart , i crossed it off my list . my last stop was the dairy aisle where i got milk and eggs . *when i had all the groceries i wanted ,* i went to the cash register ans stood in line . *when it was my turn ,* i put each item on the conveyor belt and the cashier scanned each one . a bagger put all of the groceries into my reusable bags . i paid , and then the cashier gave me a receipt . i loaded the bags of groceries into the trunk of my car and drove home .
Agenda (i.e. annotations): *evoking → take bags →get groceries →take a shopping cart →enter →irregular →check shopping list →get groceries→move along the sections →get groceries→move along the sections →get groceries→check items off the list →irregular→get groceries→irregular→wait →irregular→put stuff on the conveyor→cashier scan/weigh items →pack→pay→get receipt →bring to vehicle→leave*

Table 1: Grocery shopping stories generated by different models, together with the agendas. The text corresponding to irregular events are italicized. We could see that the text produced by our model provides much richer details than the automatic baselines.

[1]For more details and interest in reproducing our results, the data and code could be found at `https://github.com/coli-saar/Story-Generation-with-Rich-Details`.

Experiment Design We evaluate the output text by crowd-sourcing. Our evaluation has two purposes. **(1)** As our main objective, we assess how much detail a story includes, i.e. how informative a story is about a specific experience. This is captured with an *informativeness* score.
(2) We want to make sure the improvement in informativeness does not sacrifice the stories' global coherence. Therefore, we assess whether a story is globally coherent wrt. the daily activity it is about, e.g. *going grocery shopping*. That means, a story should incorporate the common sense knowledge of *going grocery shopping*, including the necessary steps and their temporal order. This part involves five questions: *syntax* evaluates the basic syntax; *global coherence* evaluates the common sense knowledge about the activity included in the story; *coverage* evaluates whether the story realizes each event in the agenda (for a human-authored story, we take its event annotations as its agenda); *relevance* evaluates whether the story stays on-topic; *local coherence* evaluates the flow of successive sentences, in terms of both content and fluency. It also verifies whether the transition between two different decoders is smooth. Ideally, our model should establish as much global coherence as Zhai et al. (2019).

The evaluation experiment is implemented with LingoTurk (Pusse et al. (2016)) and conducted on Prolific (`https://www.prolific.co/`). The questions are presented as slide-bars. We evaluate 4 stories per scenario per system, and hire 10 native English speakers to score each story.

	Coverage	Syntax	Global Coherence	Local Coherence	Relevance	Informativeness
HUMAN AUTHOR	76%^{fsz}	.62^{fsz}	.66^{fsz}	.68^{fsz}	.60^{fsz}	**.66**^{fsz}
FULL	63%^{s}	.43^{s}	.49	.42	.59	**.51**^{z}
SINGLE DECODER	57%	.38	.42	.38	.52	.47^{z}
ZHAI ET AL. (2019)	66%^{fs}	.45^{s}	.40	.44^{s}	.54	.38

^{f},^{s} ,^{z}: improvement over the respective system is statistically significant due to paired T-test at $\alpha = 0.05$.

Table 2: Results from human evaluation. The range are all $[0, 1]$.

Results The results are given in Table 2. First of all, we could see that on almost all metrics, HUMAN AUTHOR outperformed all other systems by a large margin, fulfilling its role as an upper bound. For our own model, we see that both requirements are met to validate its effectiveness: (1) it outperformed ZHAI ET AL. (2019) in *informativeness* significantly and by a large margin, indicating that the generated stories include much richer details about the specific experiences; (2) it performed comparably to ZHAI ET AL. (2019) on the first five metrics, which means it still contains the common sense knowledge of the respective daily activities, thus does not harm global coherence of the stories when trying to improve their informativeness. We also see that FULL outperformed SINGLE DECODER on all metrics, supporting our point that story writing decomposes into outlining and detailing, and that the modelling of the components should be addressed differently.

4 Conclusion

We seek to generate detail-rich stories in a coherent manner. We address the story generation process by a combined effort of two components: the *outliner* that proceeds the story, and the *detailer* that supplies context-relevant details. The system generates detail-rich stories while maintaining their global coherence.

Acknowledgement

The research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102. We thank our reviewers for their constructive feedback.

References

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara Martin, and Mark Riedl. 2019. Guided neural language generation for automated storytelling. In *Proceedings of the Second Workshop on Storytelling*, pages 46–55.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Elizabeth Clark, Yangfeng Ji, and Noah A Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2250–2260.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 329–339.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California, June. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. 2016. Inscript: Narrative texts annotated with script information. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3485–3493.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Florian Pusse, Asad Sayeed, and Vera Demberg. 2016. Lingoturk: managing crowdsourced tasks for psycholinguistics. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 57–61.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Fangzhou Zhai, Vera Demberg, Pavel Shkadzko, Wei Shi, and Asad Sayeed. 2019. A hybrid model for globally coherent story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 34–45.