Story Co-telling Dialogue Generation based on Multi-Agent Reinforcement Learning and Story Hightlights

Yu-Kai Lee and **Chia-Hui Chang**
Department of Computer Science & Information Engineering,
National Central University

Abstract

Retelling a story is one way to develop narrative skills in students, but it may present some challenges for English as Second Language (ESL) students who are learning new stories and vocabularies at the same time. The goal of this research is to develop a dialogue module for story co-telling for ESL students in order to help students to co-narrate an English story and enhance their narrative skills. However, story co-telling is a relatively underexplored and novel task. In order to understand the story content and select the right plot to continue the story co-telling based on the current dialogue, we utilize open domain information extraction techniques to construct a knowledge graph, and adopt multi-agent reinforcement learning methods to train two agents to select relevant facts from the knowledge graph and generate responses, jointly accomplishing the task of story co-telling. Compared to models that reply on chronological order, our model improves the performance from 67.01% to 70.81% through self-training with reward evaluation, achieving an increase of approximately 3.8%.

1 Introduction

Story retelling is one of the methods to enhance students' narrative abilities. However, due to weaker language proficiency, difficulty in organizing complex plots, or encountering obstacles in expressing ideas and emotions, not every student can fully elaborate on a story independently. To address this issue, we propose the task of Story Co-telling based on the concept of Scaffolding Theory (Wood et al., 1976) to assist students in story retelling. The notion of Scaffolding Theory draws an analogy from construction, where temporary support is provided during building construction, and it is removed once the construction is complete or learning is mature. Similar to training wheels when learning to ride a bicycle, Story Co-telling offers necessary support to students when needed and gradually reduces assistance as their narrative skills improve.

The objective of this study is to develop a Story Co-telling dialogue module aimed at assisting ESL students in collaboratively narrating lengthy English stories to foster narrative abilities. To refine the study's focus, we constrain the dialogue module to engage only in conversations related to story co-telling, rather than purposeless chitchat. Thus, our dialogue module is designed as a Supportive Story Chatbot, which, based on the student's ongoing narrative, determines the next plot to be told, achieving the collaborative narration of the story between two participants.

Story Co-telling is a relatively less explored and novel task, distinct from common story generation tasks. While story generation concentrates on generating logical subsequent plots, story co-telling is grounded in the content of the original story. This difference necessitates a reconsideration of model design and training methods. Since story co-telling is an interactive process between two participants, we anticipate employing reinforcement learning techniques to implement the Story Co-telling module.

However, designing a story Co-telling dialogue system based on reinforcement learning presents four primary challenges. First, it can be time-consuming and costly if we would train a dialogue system through online reinforcement learning, where the system learns from actual interactions with people. Second, utilizing offline reinforcement learning requires suitable dialogue corpora for Story Co-telling, which currently do not exist, necessitating the generation of relevant datasets. Third, the efficacy of reinforcement learning models hinges on well-defined reward functions. The task of determining how to establish appropriate environmental rewards for each dialogue round constitutes a significant challenge. Finally, when dealing with long story texts, how the agents can

Proceedings of The 21st Annual Workshop of the Australasian Language Technology Association, pages 11–19
November 29 - December 1, 2023

comprehend the entire content and choose the next coherent story plot or event is a significant challenge.

Inspired by the research by Andrus et al. (2022), we develop a Story Co-telling dialogue module based on an open-domain information extraction to condense the content of lengthy story texts,and introduce Multi-Agent Reinforcement Learning (MARL) technology to enhance the coherance and relevance of the story co-telling task. MARL involves two agents making optimal responses based on dialogue history and the Knowledge Graph built on OpenIE.

We further leverage the power of large-scale language models (LLM) to design reward functions to evaluate the quality of narratives. Specifically, we can train the reward function by carefully preparing the training data: assuming that the story highlights summarized by the LLM represent good storytelling, then modifying the story highlights by removing and adding irrelevant storylines can represent poor narrative.

By using the subjects, predicates and relationships extracted by OpenIE as the agent's action set, our model can make more informed choices across different decision contexts. Through self-trained reward evaluation, we observe that our model's performance improves from 67.01% to 70.81%, a gain of approximately 3.8%, as compared to responding solely in chronological order. This improvement indicates the feasibility of our model.

2 Related Work

The application of dialogue robots in education has garnered widespread attention. Various educational practitioners hold diverse expectations for the roles and functionalities that educational robots should embody.

For instance, the education team at the University of California, Irvine developed a system named StoryBuddy that accompanies parents and children in reading stories together. During the reading process, this system integrates question-and-answer interactions to enhance parent-child engagement (Zhang et al., 2022). They introduced the FairyTaleQA dataset (Xu et al., 2022) and employed Question Answer Generation (QAG) to address the challenge of generating questions for parents. Through experiments, it was found that implementing companion-based reading through questioning and answering enhances children's comprehension when responding to questions (Xu et al., 2021).

On the other hand, Chu and Min (2021) developed a dialogue robot specifically for retelling elementary school English storybooks. This dialogue robot assists learners in retelling stories by asking questions and utilizes rule-based mechanisms to determine whether each scene has been accurately recounted. For instance, if the first scene has been correctly mentioned, the robot prompts the student to narrate subsequent scenes. If a scene hasn't been correctly mentioned, the student is asked to retell it. Through this iterative process, students are guided step by step to independently retell the entire story. The aforementioned approach demonstrates the potential of story dialogue robots in promoting parent-child interactions, cultivating reading interests, and enhancing narrative skills.

Continuing with the theme of enhancing children's narrative abilities, recent research has also focused on utilizing information extraction techniques to comprehend and analyze long-text narratives. These techniques aim to transform unstructured textual data into structured information. For instance, Xu et al. (2023) developed a Document-level Narrative Event Chain Extraction Toolkit (NECE). This approach employs technologies such as Semantic Role Labeling (SRL) to extract relevant information about characters and events from stories. Furthermore, a specific TF-IDF algorithm is used to identify the most important events. Through this framework, the narrative structure within lengthy textual stories can be effectively parsed, enabling the extraction of essential elements like characters and events.

Similarly, Andrus et al. (2022) address the challenge of understanding long-text narratives using dynamic knowledge graphs. Unlike static commonsense knowledge graphs that involve real-world information, Andrus et al. (2022) utilize OpenIE (Open Information Extraction) technology to construct dynamic knowledge graphs. These dynamic knowledge graphs are then applied to tasks such as question answering and story completion. This approach proves effective in overcoming the limitations imposed by language model input constraints when dealing with lengthy documents, and its effectiveness has been demonstrated.

3 Method

The MARL structure for story co-telling based on knowledge graph construction is shown in Figure

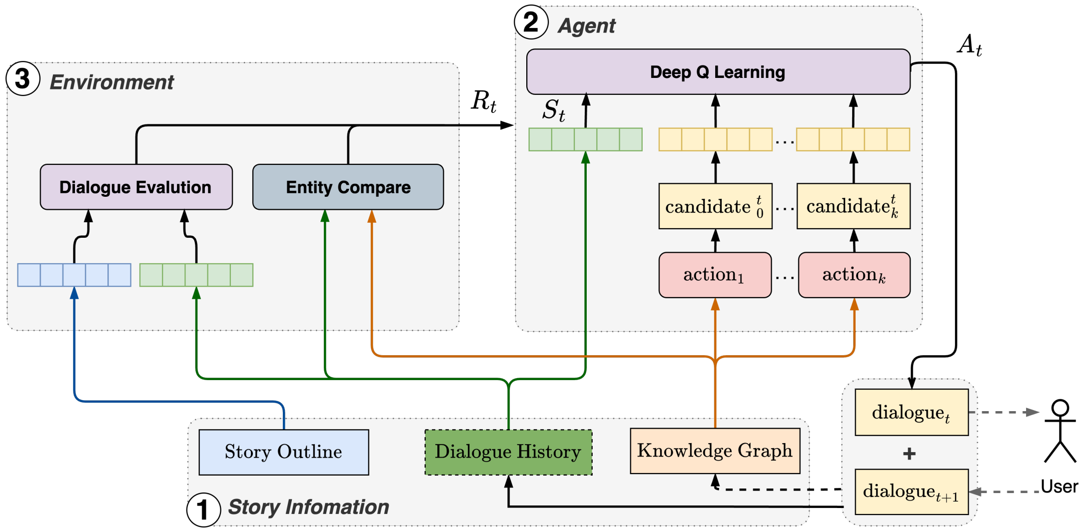

Figure 1: Architecture of the Story Co-telling Module via Reinforcement Learning and Knowledge Graph

1. We will start from how to convert a long text story into a knowledge graph and introduce how the agent uses conversation history and knowledge graph to select the plot to be told next. Secondly, we explain how to construct a dialogue history evaluation model for evaluating the current performance of story co-telling. Finally we will explain how to use reinforcement learning to integrate the above parts into a story sharing dialogue module that can make decisions based on the current dialogue history.

3.1 Long Text to Knowledge Graph

The purpose of constructing knowledge graphs is to distill information from lengthy text narratives and transform unstructured data into a structured form. This enables our model to effectively comprehend the storyline of the narrative. We utilize Stanford CoreNLP toolkit, the OpenIE (Open Information Extraction) framework (Angeli et al., 2015), version 4.5.4, to extract structured fact triples (i.e. subject, relation, and object) from text.

For example, consider the sentence "After a time there was another feast, and the Many-furred Creature begged the cook as at the last one to let her go and look on." Even though this sentence describes "the Many-furred Creature begged the cook to let her go and look on, just like the last time," due to the constraints of the triple representation, the second object, time, location, and other words need to be separately recorded. Hence, the preceding sentence can be represented by three fact triples: [many furred creature, begged, the cook], [many furred creature begged the cook, adv, as at the last one], and [many furred creature begged the cook, arg2, to let her go and look on]. These triples are then visualized as a directed graph, as depicted in Figure 2.

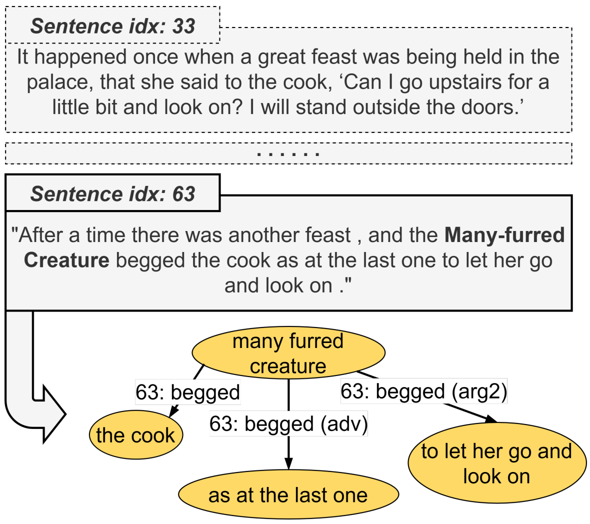

Figure 2: Example of Constructing Knowledge Graph Using OpenIE

To mitigate potential redundancy in the fact triples produced by open-domain information extraction models, we remove duplicate triples and retain longer ones to preserve more information. Additionally, we also employ Coreference Resolution (Recasens et al., 2013) to process the text and replace pronouns with the nouns they refer to.

In practice, in addition to the subject, relation, and object triples, we also record the sentence index $sidx$ of each fact triple in the original story to understand the context of the fact triples. Additionally, we also keep a status indicator for each fact to record whether it has been mentioned in the conversation. This helps prevent repeated references to the same fact during the narrative.

3.2 Agent

In this paper, we employ Deep Q-Learning for designing the conversational agent. The agent makes decisions based on the current state S_t, takes the next action A_t, and adjusts its decisions according to the feedback rewards R_t generated by the environment. Here, S_t is a vector composed of various pieces of information, including the conversation history $D = [u_0, u_1, ..., u_t]$, and candidate responses $C_t = [c_0^t, c_1^t, ..., c_k^t]$ generated by corresponding strategies $A = [a_0, a_1, ..., a_k]$. We use Sentence Transformer (Reimers and Gurevych, 2019) to convert these text fragments into vectors expressing their underlying information. After passing through Deep Q Learning, the agent selects the candidate response to be used for the reply, which determines the next action $A_t = i$, where $i \in [0, k]$. We will now introduce the action design of the agent and the methods for generating candidate responses. The details of the reinforcement learning will be discussed in the subsequent sections.

Action Design

To ensure coherence in the co-told story, the agent, based on the latest utterance u_t in the conversation history, utilizes the Sentence Transformer to find the top three relevant facts on the knowledge graph G as reference points p of the interlocutor's current narration. Subsequently, using these reference points, four distinct strategies are employed to extend the conversation, thereby generating candidate responses. Each strategy is treated as an action a_i. Here's a brief description of each action:

- a_0: Select subsequent events from the reference point. In other words, choose facts f where $f.sidx$ is greater than $p.sidx$.
- a_1: Choose facts with subjects similar to the subject of the reference point p.
- a_2: Choose facts with relations similar to the relation of the reference point p.
- a_3: Choose facts with objects similar to the object of the reference point p.
- a_4: Declare the end.

Response Generation

We can utilize the story sentences, along with their corresponding fact triples obtained using OpenIE, to prepare training data for T5 model training, i.e. create an input-output mapping using the facts triples and story outline as input and the sentence that contributes the fact triples as output. By fine-tuning, we enable the T5 model to generate results similar to the original sentences based on the given fact triples and the story outline. An example input format is depicted in Figure 3.

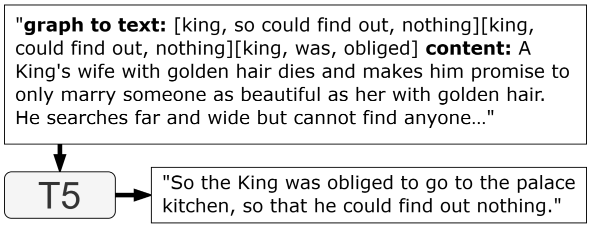

Figure 3: Fine-tuning the T5 Model for Knowledge Graph to Text Generation

3.3 Environment: Reward Function Design

The reward function is mainly divided into two parts: dialogue history assessment and entity connection assessment. The former provides an overall rating of the dialogue up to the current point, while the latter calculates the connection rating between the current turn and the previous sentence.

Dialogue History Assessment

To evaluate the effectiveness of the co-told story dialogue history D, we require both positive and negative co-telling examples along with their ratings. These examples can be used to train a regression model for automatically assessing the quality of co-told stories.

Due to the lack of readily available co-telling dialogue datasets, we utilize ChatGPT to generate a specified number of bullet-pointed story highlights for each story. As shown in Table 1, we design a prompt to guide ChatGPT in generating the desired number of story highlights H for the story text. To facilitate further processing, the generated results are output in JSON format. Considering ChatGPT's generation diversity, the same prompt can lead to various outcomes.

We generated story highlights using ChatGPT and subsequently performed actions such as replacement or deletion to create lower-quality story highlights. This approach of generating story high-

Input
<|Plots|> = number of plots that you want to generate <Story_text> = story corpus
Prompt
Please summarize the following Story by outlining <$\|Plots\|$> plot points in JSON format in order. (example: [{“plot_id": 1, “plot_point": first plot point}, {“id": 2, “plot_point": second plot point}]) Do not provide additional information or comment. – Story: <Story_text>

Table 1: Prompt Format for Generating Story Highlights Using ChatGPT

lights can be seen as generating poor examples in co-telling, as they may disrupt the integrity and logic of the story. Depending on the number of replacements or deletions, we assign different scores.

As the impact of replacement and deletion on the quality of story highlights differs, we have formulated separate adjustment formulas and evaluation formulas for these two actions. The formula for deleting story highlights is presented in Eq.(1), while the formula for replacing story highlights is shown in Eq.(2).

$$score = e^{\left(-1.6\times\frac{n}{|Plots|}\right)} \times 9 + 1 \qquad (1)$$

$$score = e^{\left(-4\times\frac{n}{|Plots|}\right)} \times 10 + 1 \qquad (2)$$

Here, n represents the number of modifications, and $|Plots|$ represents the original number of story highlights. We believe that replacing an existing story highlight with another storyline has a greater impact on the overall quality compared to deleting a single story highlight. As a result, replacing a larger number of story highlights will receive a lower score compared to deleting the same number of story highlights (see Figure 4).

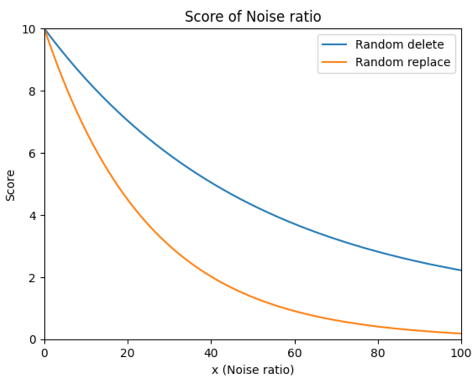

Figure 4: Score vs. # of edit operations

Dialogue history assessment is essentially a regression problem, as illustrated in Figure 5. We input both the dialogue history D and the story outline H into the same RoBERTa (Liu et al., 2019) model, and extract the hidden state of the CLS token from the model. Subsequently, the two hidden states are concatenated and fed into a neural network. This network outputs a score $DH(D, H)$ between 0 and 10 to evaluate the quality of the co-told story.

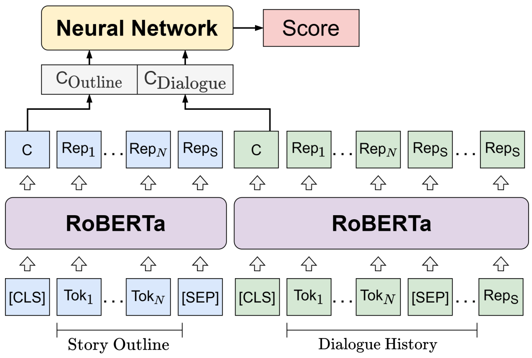

Figure 5: Architecture of the Dialogue History Assessment Model

Entity Relationship Evaluation

The purpose of entity relationship evaluation is to assess whether the current reply (R) is related to the entities (E) mentioned in the previous sentence of the story. We utilize OpenIE to parse these two sentences and employ BFS graph algorithm to determine if these two entities can be connected in the knowledge graph. If the two entities are linkable in the knowledge graph, we consider there is an entity relationship between these two sentences and provide quantitative rewards as feedback.

We compute the score $DH_t = DH(D_t, H)$ for dialogue history assessment and the entity connection assessment score $EC_t = EC(R_t, E_{t-1})$ for each round t. Since dialog history assessment is an accumulated score, we thus take the score difference of two subsequent rounds along with the entity connection score as the reward R_t for this round as indicated in Eq. (3). This reward is subsequently fed back to the agent.

$$R_t = DH_{t-1} - DH_t + EC_t \qquad (3)$$

3.4 Multi-Agent Reinforcement Learning

Finally, we apply Deep Q Learning (DQL) and Multi-Agent Reinforcement Learning (MARL) methods to enable two agents to collaboratively perform the task of co-telling a story (see Figure 1). Through the guidance of reward scores, the agents

turn	history	score
...	...	...
6	The Princess falls asleep in a hollow tree and is discovered by the King's huntsmen.	7.37
7	The King's huntsmen bring the Princess to the palace and she is assigned to work in the kitchen as the Many-furred Creature.	7.34
8	The Many-furred Creature lives in poverty and works in the kitchen doing all the dirty work.	7.79
9	The Many-furred Creature attends a feast at the palace and enchants the King with her beauty.	7.95
10	The Many-furred Creature cooks soup for the King and hides a gold ring in it.	8.06
...	...	...
14	The King and the Princess live happily ever after.	8.02
	Final Score=8.02, Gold=9.09	

turn	history	score
...	...	...
6	The Princess falls asleep in a hollow tree and is discovered by the King's huntsmen.	7.37
7	**The Emperor takes Confucius' shoes and staff as a joke, but the tablet's warning comes true and he dies soon after.**	**6.34**
8	**The cock gets the garland and trades it for red silk from the brook.**	**4.82**
9	**The jackdaws and magpie eat the leftover pie-crust and gravy.**	**2.74**
10	The Many-furred Creature cooks soup for the King and hides a gold ring in it.	2.17
...	...	...
14	The King and the Princess live happily ever after.	6.21
	Final Score=6.21, Gold=7.38	

Table 2: Examples of conversation history evaluation model. The left table shows high-quality storyline highlights (which received a score of 9), while the right table shows cases where the inclusion of irrelevant content resulted in a drop in reward points.

learn how to continue the story. While the story co-telling agent will only focuses on a single story during the interaction with the user, updating the model based on a single story is dangeous because the model is likely to forget what it has learned in the past. Therefore, We choose to adopt experience replay mechanism to avoid catastrophic forgetting.

Our objective is to enable two agents to collaboratively co-tell a story. In each dialogue turn, the agents take turns transmitting the selected response through the environment, without sharing their respective knowledge graph states. This implies that each agent can only understand the co-told story and make appropriate responses based on the co-telling conversation history. If one of the agents terminates prematurely, the entire dialogue also ends, followed by subsequent analysis and evaluation. This design simulates real-world human-machine interaction scenarios, challenging the agents' understanding and response decision-making abilities.

Before the training begins, we will initialize each environment and model (lines 1 to 5). In each epoch (line 6), we engage in a dialogue for each story (line 7), simultaneously initializing the environment state before the co-telling begins (lines 8 to 11). In lines 12 to 23, it can be observed that the two agents take turns generating candidate responses, connecting their vectors with the dialogue history vector to form the current state representation (lines 13 to 14). Subsequently, the agents use their own Q Network to decide which candidate response to select (lines 15 to 16). Following this, we employ the Dialogue Evaluation Model and Entity Compare to generate rewards (lines 17 to 19), while also producing the next state (lines 20 to 21). Finally, the tuples of state transition, action, next state, and corresponding reward (s, a, s_{t+1}, r_{t+1}) are stored in their respective memories (line 22), for subsequent learning and updating processes.

4 Experiment

In this study, we chose stories from FairytaleQA (Xu et al., 2022) as the designated story set for story co-telling. These stories are classic fairy tales suitable for readers below the ninth grade, with clear narrative structures. The average text length of stories used in FairytaleQA exceeds one thousand words. Additionally, with the pre-designed question-answer pairs available in FairytaleQA, we can evaluate the diversity of co-telling content through question answering.

To ensure the effectiveness of agent training, we set some termination conditions for the environment. Firstly, by limiting the conversation rounds to be no more than 20, we avoid resource wastage and increased training time caused by excessively lengthy dialogues. Additionally, when one of the participants introduces an ending keyword, it signifies an appropriate endpoint for the conversation. Furthermore, we set the exhaustion of all facts in the knowledge graph as one of the ending conditions. This configuration ensures efficient utilization of information during the conversation and prevents the repetitive use of the same facts.

Algorithm 1: Story Co-telling MARL

Data:
$I = [(O_1, G_1), (O_2, G_2), ...]$ Story info.;
O_j = Story outline;
G_j = Story knowledge graph;
Function:
μ = State embedding model;
Φ = Candidate response generate func.;
Θ = Dialogue evaluation model;
Ξ = Entity compare func.;
Training:
Initialize $Agnet1$ and $Agnet2$;
Initialize Q Network Q_1 and Q_2;
Initialize epsilon ε;
Initialize replay memory M_1 and M_2;
Initialize environment E_1 and E_2;
foreach *epoch* **do**
  foreach (O_j, G_j) *in* I **do**
    Reset dialogue history D;
    Reset environment E_1 and E_2 by (O_j, G_j);
    $t = 1$;
    $Score_t = 0$;
    while *(*E_1 *is not done) and (*E_2 *is not done)* **do**
      $C_t \leftarrow \Phi(D, G)$;
      $s_t \leftarrow \{\mu(D), \mu(C_t)\}$;
      $a_t \leftarrow argmax(Q_{t\%2}(s_t, \varepsilon))$;
      $d_t \leftarrow C_t[a_t]$;
      Append d_t to D;
      $Score_{t+1} \leftarrow \Theta(O_j, D) + \Xi(G_j, D)$;
      $r_{t+1} \leftarrow Score_{t+1} - Score_t$;
      $C_{t+1} \leftarrow \Phi(G)$;
      $s_{t+1} \leftarrow \{\mu(D), \mu(C_{t+1})\}$;
      Append (s, a, s_{t+1}, r_{t+1}) to $M_{t\%2}$;
      $t = t + 1$;
    end
    Update Q_1 by M_1;
    Update Q_2 by M_2;
  end
  Update ε;
end

4.1 Dialogue History Evaluation Model

During the training of the dialogue history evaluation model, we set the batch size to 1 and conducted 20 training epochs. Across these training sessions, the loss value on our training set was 0.0197, indicating a strong fit of the model to the training data (Figure 6). The best validation set loss was 0.0299, demonstrating satisfactory performance on unseen data. Additionally, we computed the Pearson correlation coefficient between the scoring values and the dialogue history evaluation model, yielding a value of 0.8313, indicating a positive correlation between the data labels (given by Eq. (1), (2)) and the model's outputs.

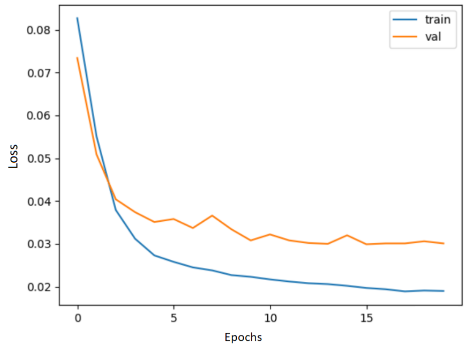

Figure 6: Training of the Dialog History Assessment Model in Figure 5

Table 2 presents the scoring results provided by the dialogue history evaluation model on two conversation history examples. The "score" column displays the cumulative score from the first utterance up to the current turn, and the gold score for the entire conversation are marked at the bottom. As shown in the example, when the input contains high-quality story focus, the model's output results closely match the default scores. This indicates that our dialogue history evaluation model can accurately assess story focus and assign appropriate scores. If irrelevant story focus is inserted into the story, the scores given by the dialogue history evaluation model significantly decrease. This further demonstrates the effectiveness and feasibility of our dialogue history evaluation model, as it can identify relevant story focus and provide appropriate evaluations for them.

4.2 Effectiveness of Story Co-Telling Models

Secondly, we conducted a performance comparison with rule-based responses, which involves responding solely based on chronological order, i.e. a_0. Figure 7 illustrates our training results, demonstrating that both single-environment reinforcement learning (1Env) and multi-environment reinforcement learning (2Env) outperform the rule-based responses. The performance of multi-environment reinforcement learning is the best. According to the feedback values from our trained dialogue history evaluation model, the performance of multi-environment reinforcement learning has improved by approximately 3.8%, from 67.01% to 70.81%, compared to responses based on chronological order, i.e. choose action a_0.

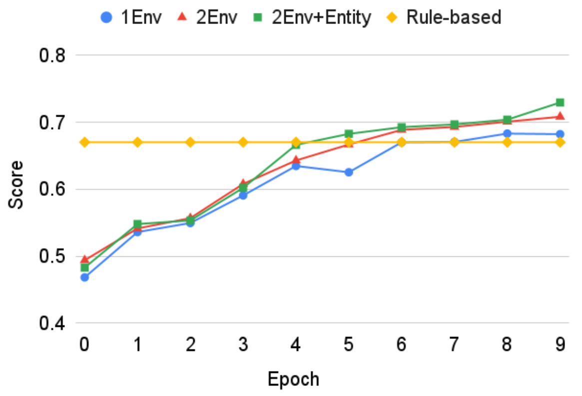

Figure 7: Comparison of Results from Story Co-Telling Trained with Different Methods

This result indicates the feasibility of multi-agent reinforcement learning methods in the story co-telling task. Compared to rule-based responses that rely solely on chronological order, our model, trained through the interaction of multiple agents, can better comprehend dialogue history and generate responses based on the knowledge graph. This enables our model to provide more coherent and relevant replies, further enhancing the quality and experience of the conversation.

4.3 Comparison of Reward Function Design

Next, we investigate the effect of incorporating entity connection EC reward on the model's action selection. As shown in Figure 8, we can observe that both the average EC and DH reward increase over the course of training. Furthermore, in comparison to using only the dialogue history evaluation model as the sole reward (DialogueEvaluation), under the encouragement of entity relationship evaluation (DialogueEvaluation + EntityCompare), the model tends to choose actions related to entities (as shown in Figure 9). This indicates that the approach of introducing entity comparison into the dialogue history evaluation model has a certain impact on the model's decision-making process.

4.4 Discussion: Evaluation of Co-told Stories

Finally, we try to evaluate whether the co-told stories are good or bad. One possible way is to use question answering to test whether the story hightlights can answer the pre-designed questions. We conducted experiments using a fine-tuned T5 question-answering model (Christian Di Maio, 2022) based on the story summaries. We replaced the story paragraphs corresponding to questions in FairytaleQA with the story summaries to evaluate whether the story summaries could effectively answer questions from the stories.

The experimental results are presented in Table 3. The performance of this fine-tuned T5 model on story summaries is not ideal. This is mainly because the story summaries are relatively short, lacking details and context, which makes it difficult for the question-answering model to provide accurate answers. Additionally, the story summaries might contain implicit information, requiring the model to possess stronger reasoning abilities to handle such implied content.

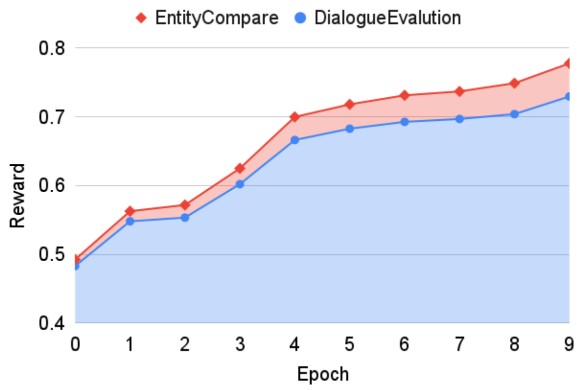

Figure 8: Stacked Area Chart of Entity Relationship Reward during Training Process

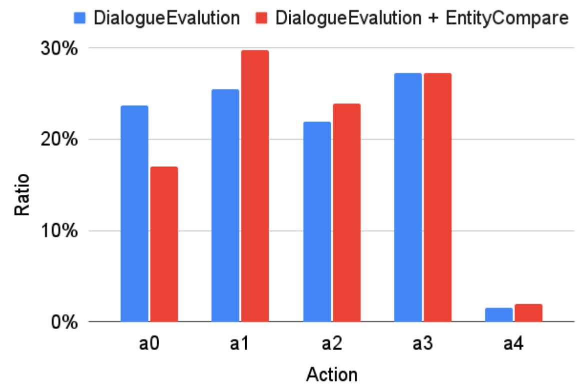

Figure 9: Change of Action Selection (in Section 3.2) Histogram

Question Types	Train		Val		Test	
	F1	EM	F1	EM	F1	EM
character	**24.11**	**16.53**	**27.33**	**18.69**	**20.41**	**11.65**
action	11.85	2.19	13.64	3.00	13.27	2.54
setting	15.50	6.50	23.64	6.67	14.34	3.23
feeling	4.60	3.28	3.26	1.06	7.97	4.72
causal relationship	15.87	0.12	17.19	0.00	19.10	0.36
outcome resolution	12.18	0.12	14.22	1.03	17.39	0.00
prediction	16.34	3.55	19.23	1.82	16.30	0.00
All	**14.09**	**3.46**	**15.93**	**3.51**	**15.63**	**2.78**

Table 3: Performance of Fine-Tuned T5 Model on FairytaleQA under Story Summaries

5 Conclusion and Future Work

In this study, we designed a dialogue module for story co-telling with the aim of enhancing ESL students' English narrative abilities. By training two agents to select optimal responses from the knowledge graph based on dialogue history, our model is capable of making wiser choices among candidate responses generated by different decision actions. Through self-training reward evaluation, we observed that our model's performance improved from 67.01% to 70.81% compared to responding based solely on chronological order.

For future work, the knowledge graph is still limited by the completeness and coverage of openIE performance. Therefore, we can try chatGPT to enhance information extraction. Furthermore, while our current approach centers on action design guided by coherence, alternative strategies, such as considering story coverage, could also be employed to shape these actions. Overall, there is still a lot of room for improvement in this research.

References

Berkeley R Andrus, Yeganeh Nasiri, Shilong Cui, Benjamin Cullen, and Nancy Fulda. 2022. Enhanced story comprehension for large language models through dynamic document-based knowledge graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10436–10444.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Giacomo Nunziati Christian Di Maio. 2022. Mariorossi/t5-base-finetuned-question-answering (huggingface). `https://huggingface.co/MaRiOrOsSi/t5-base-finetuned-question-answering`.

Seong Yeub Chu and Deok Gi. Min. 2021. Development of an ai chatbot-based teaching model for english picture book retelling activities. *Modern English Education*, 22(4):37–50.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633, Atlanta, Georgia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

David Wood, Jerome S Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, 17(2):89–100.

Guangxuan Xu, Paulina Toro Isaza, Moshi Li, Akintoye Oloko, Bingsheng Yao, Cassia Sanctos, Aminat Adebiyi, Yufang Hou, Nanyun Peng, and Dakuo Wang. 2023. Nece: Narrative event chain extraction toolkit.

Ying Xu, Dakuo Wang, Penelope Collins, Hyelim Lee, and Mark Warschauer. 2021. Same benefits, different communication patterns: Comparing children's reading with a conversational agent vs. a human partner. *Computers & Education*, 161:104059.

Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 447–460, Dublin, Ireland. Association for Computational Linguistics.

Zheng Zhang, Ying Xu, Yanhao Wang, Bingsheng Yao, Daniel Ritchie, Tongshuang Wu, Mo Yu, Dakuo Wang, and Toby Jia-Jun Li. 2022. Storybuddy: A human-ai collaborative chatbot for parent-child interactive storytelling with flexible parental involvement. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.