School of Informatics

Informatics Project Proposal
Collaborative Storytelling with Continuation Recommendation

[redacted]

May 2021

Abstract

Previous work on automatic storytelling largely focuses on generating a complete story in one go based on limited user-provided input, where users only have little control over how the story unfolds. In this project, we propose the task of story generation with continuation recommendation to enhance user controllability and system predictability. Thus, given a beginning sentence of a story, the task is to offer recommendations on the next sentence for user selection. We will use existing story generation models to generate sentence candidates and then explore the heuristics to evaluate each candidate and make recommendations.

Date: Monday 17^{th} May, 2021

[redacted]

1 Motivation

Recent technical advancements on text generation models [1][2] attracted attention from researchers to the field of automatic storytelling. Automatic storytelling is a form of digital interactive experience, where users collaborate with artificial intelligence to create or influence a story through actions like issuing the command to the system [3]. It has been applied in many fields [4], ranging from entertainment, such as interactive movies and computer games, to serious applications like education and therapy. Existing research in automatic story generation focuses on different approaches to solve the controllability problem: the user input's ability to influence the generated results [5]. Yao et al. [6] and Rashkin et al. [7] attempts to control the story by keyphrase-based storylines. Amanabrolou et al. [8] and Wang et al. [9] proposed ending-guided methods, where intermediate sentences are generated based on a specific desired starting and ending. Although these approaches have some potential interactions that enable the writer to control the story somehow, there is no research on recommending alternative sentences to continue a story that has been written so far.

1.1 Problem Statement

In this project, the aim is to fill the gap in recommending story continuations to the users in automatic story generation. We formulate it as a recommendation system problem, where each sentence candidate is an item, and a set of them defines the item space from which recommendations are made. We propose to decompose story generation into two processes. One is the generation of alternative story continuation sentences with a given input. The other is a continuation recommendation process built on top of the sentence generation process to create a list of recommendations on story continuation with considerations of user preferences. In this way, given a starting or/and ending sentence of a story from the users, the system firstly generates a complete set of candidates and then offers recommendations as per some criteria. The users could pick one of them based on their intentions to continue the story. We will explore how to build a framework for integrating these two processes, specifically exploring how to generate different candidate sentences, the heuristics to evaluate each sentence, and the rules to make recommendations.

1.2 Research Hypothesis and Objectives

The main aim of the project is to answer the following research question: How can we best set up a recommender system for story continuation from automatically generated sentences?

To achieve that, we propose to decompose the recommender system problem into generating the item space and selecting items to recommend. We will firstly investigate how to use the existing story generation models to generate different candidate sentences, then, explore different approaches to select and recommend story continuation, such as sentiment analysis, emotion classification, clustering, and techniques in the field of recommendation system. Finally, we will conduct both quantitative analysis and user study to evaluate the results from different approaches and find the best solution.

1.3 Timeliness and Novelty

The proposed research is conducted at the most suitable time frame. It can take advantage of the last development of powerful transfer-based pre-trained language models, such as GPT-2 [1] published in 2019, to generate more coherent stories. Furthermore, our system could build on top of the recent work in automatic story generation and extend the work to solve our recommender system problem. We have studied the related work and identify the potential interactions between system and users in the literature. However, no research focuses on recommending story continuations to the users with consideration of their preference. Therefore, our research is of great novelty as we are creating a new paradigm in automatic story generation.

1.4 Significance

Recent state-of-the-art models [6][7][9] in this domain generate a complete story in one go based on limited user-provided input, where users only have little control over how the story unfolds. The generated story is deterministic but has random content without considering user preferences. It is learned that the users, such as the artists and writers, require elements of control and predictability in their automatic tools to assist their work effectively as expected [10]. Motivated by this, we create a new paradigm that frames automatic story generation into two processes, aiming to recommend various story continuations to the users or even incrementally learn from the user selections to offer better recommendations. We further formulate the task as a recommender system problem. The proposed work fills the gap in recommending sentence continuation, driving the improvements in automatic story generation. It provides desirable options for user selection, giving them more freedom to choose the directions of the story based on their intentions, hence improving their interactive experience with intelligent systems.

1.5 Feasibility

The proposed pipeline of storytelling with recommendation consists of sentence generation and continuation recommendation processes. We have identified several alternative machine learning models in the related work and methodology section. Either pre-trained models or code implementations of [1][7][6] are publicly available. They can be used to create our main pipeline without the need to build and train the model from scratch. Many heuristics can be explored in the continuation recommendation processes and thus produce several pipelines for our system. We will explore one to three of them depend on our progress to ensure the workload is appropriate and the work could be completed within the 10-week time frame. All of these suggests that the project can be completed in the due time.

1.6 Beneficiaries

The proposed work will benefit the users of the system and the research field of automatic story generation. The system will serve as a tool to help a wide range of users, such as the writers or anyone who is interested in storytelling, to write the story, book, or even movie script automatically while with more control. The recommendations made by the system might provide the user inspirations in writing. Besides, we identify the exiting interactive elements in the literature and propose several ways to generate alternative sentences, which provides other researchers with a way to track the current progress in the literature and allows them to continue the work in this scope. Our proposed pipeline could inspire further innovations in story continuation recommendation.

2 Background and Related Work

2.1 Automatic story generation

Existing work has proposed several approaches for automatic story generation. We divide them into two categories according to controllability, which is user input's ability to influence the generation results. We focus on their methodology, strengths, and weaknesses, followed by identifying potential interactive elements for writing stories.

Storyline-focused models Storyline-focused models generate stories from an outline of the plot. Yao et al. [6] proposed a plan-and-write generation framework where story generation is decomposed to story planning and surface realization. The framework uses the story's title to generate a storyline that consists of a sequence of keywords and then takes them as input to the sequence to sequence model for controlling the story generation. The experimental results show that storyline-conditioned generation improves the coherence and diversity of generated story compared to those baselines with simple title inputs in a generation. While, the system still suffers from several typical problems in story generation, such as repetitive sentences, off-topic plot, and logically inconsistent story. Along the line, Rashkin et al. [7] introduced a novel transformer-based narrative model built on top of GPT-2 [1] that generates a multi-paragraph story conditioned on the provided plot outline represented as an unordered list of key phrases. They employed a memory mechanism to track the dynamic plot states computed using outline and story generated so far. The model takes the memory state, the representation of the previous paragraph, outline, and discourse as input to generate a whole paragraph at each step. The quantitative evaluation shows that plot state tracking contributes to the coherence and consistency of the composed story. These storyline-guided models provide users more fine-grained control to the whole story, while their performance is degraded with inappropriate choices of key phrases in a storyline.

Beginning or/and Ending-focused Models The second category of research focuses on controlling a story by specifying the story's starting or/and ending. In light of challenges in a storyline-focused generation, Wang et al. [9] proposed a generation-by-interpolation story generation model that incrementally takes a sentence pair as input and generates the intermediate sentences to connect them. The process starts with the given beginning and ending sentence of the story. At each step of sentence generation, the model uses GPT-2 [1] to generate a list of interpolation candidates based on context pair, and then implements a coherence ranker with pre-trained RoBERTa [11] to evaluate the coherence of each candidate before selecting the globally optimal sentence as output. The results demonstrate that the ending-guided approach with a coherence ranking evaluation technique helps generate more coherent and faithful stories than storyline-guided approaches. However, the model is bad at commonsense inference, implying the need for human knowledge injection. Xu et al. [12] introduced a new model called SoCP that generates a story with multi-character and multi-emotion control. They used a sequence to sequence model for generation based on the seed beginning sentence and adopt character selector character and emotion controller in the decoder to help the sentence generation with specified emotion for each character. The experiment results demonstrate the effectiveness of their approach in story generation with assigned input beginning and emotion changes, promoting controllability. Brahman et al. [13] attempt to control the generation process dynamically through the user-provided beginning and mid-level sentence abstractions. Unlike typical one-shot story generation, this interactive model generates one sentence at a time, and the user provides a cue phrase to guide the content of the next sentence. It consists of two separate encoders to encode context and cue phrases and the attention mechanism to ensure the output sentence are semantically related to the context and cue phrase. Both automatic and human evaluations show that the model produces more topically coherent and personalized stories compared to the plan-and-write model [6] in the interactive setting.

Interactive Elements for Writing Stories It has been studied that most people love collaborating with machines [14]. Although previous work has mainly focused on generating a story in one go, there are some potential element of choices in the system that could be provided to the user to decide how the story unfolds. The dynamic model developed by Yao and collaborators[6] uses the user-provided title and the previous generated sentence to produce part of the storyline that guide the next sentence generation. Instead of an auto-generated story plan and one-shot story generation, we could design the model to take each key phrase in the storyline as input separately and generate the story one sentence at a time. In this way, the user could control what happens next interactively during the generation process based on the story generated so far. Similarly, Brahman et al. [13] propose that the user can give different cue phrases to guide the system in generating the next sentence after providing the beginning sentence of the story. Xu et al. [12] present a story generation system that can generate stories with appointed emotional changes for each character according to psychological theories. We can take these emotions in psychological theories as a list of options for the users to choose for each character. In the research [9], a coherence ranker is used to pick out the globally best intermediate sentence for the story. Instead, we could make the system show the users the top-k highest coherent sentences and let the user pick, enabling a more interactive and controllable story generation. Roemmele et. al [15] developed an application called creative help for writing stories, where the users can request a suggestion for the next sentence in a story in the course of writing. The generation model accesses a large corpus of stories to find a related next sentence, which the user could modify.

2.2 Recommendation system

A recommendation system is an information filtering system that filters important information fragment from a large amount of dynamically generated data to solve information overload [16][17]. It can predict the user preference on an item and provide suggestions to the user [18]. The suggestions involve various decision-making processes, which is what sentence to choose as story continuation in our case. In order to make recommendations, the system will firstly query users preferences and then decide which items to recommend. There are several basic approaches for item selection [19]. Collaborative filtering approach assumes users will share the same interests in the future if they do so in the past. So, if two users have strongly overlapped interests and one of the user has recently selected an item that is unseen to another user, the system will propose the same item to him/her. Content-based filtering approach recommends items that are similar to those the user like in the past. It uses similarity-based retrieval or text classification methods to measure and select items. Knowledge-based approach exploits additional and means-end knowledge about the available items and the user and makes recommendations by calculating similarities between customer requirements and items or relying on explicit recommendation rules.

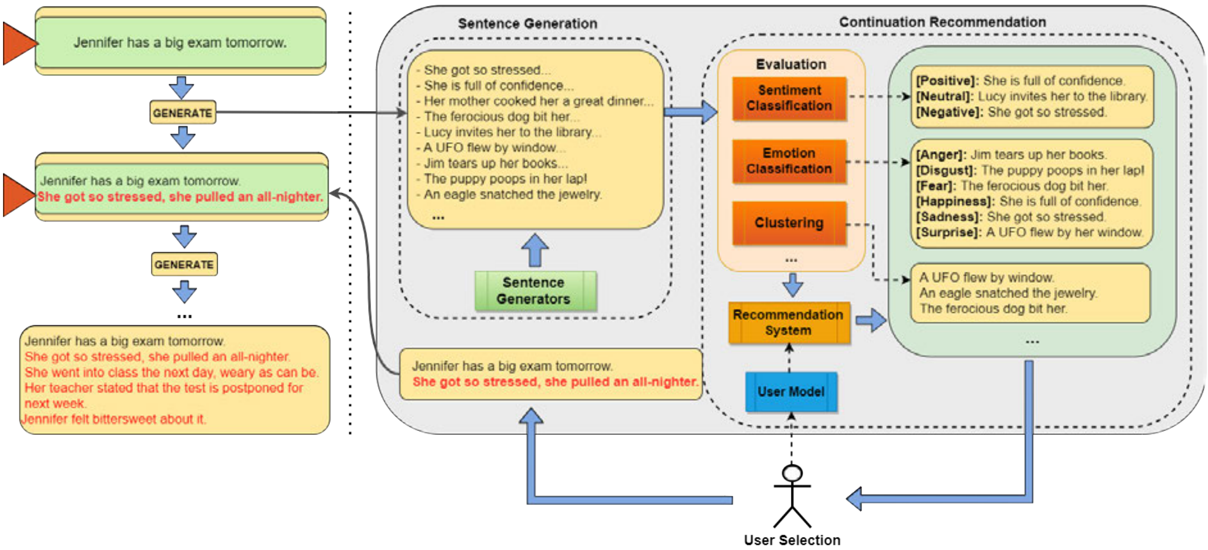

Figure 1: System Pipeline.

3 Programme and Methodology

We propose a novel storytelling system with story continuation recommendations to help the user write the story. Given a input sentence, the system produces recommendations on the next sentence for user selection to continue the story and the procedure runs iteratively. We define this as a recommender system problem. To address it, we decompose the interactive story generation into two main processes, namely sentences generation and story continuation recommendation. Fig. 1 shows the whole pipeline of the storytelling system. For a user-provided beginning sentence, the system has sentence generators that generate multiple candidates of next sentence, defining item space for recommendations. Then, there is an evaluation process that evaluates each candidate in terms of sentiment, emotion, grouping or other criteria and produces a vector representation describing the attributes of each item. With these representations, the recommender system picks a set of items based on specific rules and present to the user. After the user selects a preferred sentence for the story continuation, the selected sentence takes the selected sentence and the previous sentences as input for the next iteration.

In actual system implementation, we will build the **first system pipeline** that incorporates one sentence generator (Section 3.1), one evaluation technique (Section 3.2), and one basic rule for the recommendation (Section 3.2) in Fig. 1. After that, we will explore more heuristics to develop our **second and third pipelines**. To better explain the methodology, we arrange the following subsections in terms of system processes. In each subsections, we will identify the challenges and describe how we plan to implement each of the critical processes.

3.1 Sentence Generation

Generating different candidates of next sentence is crucial as it composes the item space for the subsequent recommendation, while it is also challenging since the existing story generation models only generate a single best sentence for story continuation based on their objectives.

To solve this, we propose four potential ways to generate different continuation sentences for a given input. We can use one or more story generation models discussed in the related work section as the baseline models for the sentence generation process. The first way is to select a non-deterministic model, where the output sentence for the same input is different in every run. We then feed the same sentence to the model as input and run it at different go to acquire different alternative sentences. Since not all models are non-deterministic and always generate the same output for the same input, the second way is instead to select several deterministic models and provide them with the same input to form a set of alternative sentences, each from one of them. Alternatively, we could retain one or more deterministic models or finetune them at different degrees or with different datasets. Then, we take each instance of the model to generate different continuation sentences. The fourth approach is to transform the input data but still maintain the central semantics of them by paraphrasing. In this way, we could have several paraphrases for the input sentence for the models to generate different outputs.

In the actual implementation, we will select one of them with considerations of source availability, such as pre-trained models, codes, and computational resource.

3.2 Continuation Recommendation

Sentence Evaluation

After defining the item space, a complete set of sentence candidates, the system evaluates these items for the recommendation based on criteria using machine learning techniques. It generates a set of attributes associated with each item in vector representation. In the context of storytelling, one of the important elements of a story is the changing tone as the story unfolds, indicating particular feelings of a story and reflecting the writer's attitude [20]. Therefore, sentiment analysis could be used to detect the attitudes contained in the sentences as per Scherer's [21] typology of affective states. It classifies each candidate into one of three sentiment polarities: positive, negative and neutral. We will employ one of the successful sentence-level sentiment classifiers to assign the sentiment attribute to each candidate. Alongside this one-dimensional view of tone, we could develop the system to perform an emotion evaluation of an event presented in each sentence to capture the degree of all six basic emotions (i.e. anger, disgust, fear, joy, sadness, surprise) [22]. Furthermore, we could also represent each sentence as sentence embedding using a pre-trained text encoder like CLIP [23] and then perform clustering over the item embedding space. In this way, we group semantically similar candidates into several clusters and measure the distance from the sentence to the centroid of each cluster. The cluster ID and distance serve as attributes, helping identify the representative sentences and the most different sentences (i.e. the outliers) in the item space. There could be other attributes, and all these attributes will be represented by a vector to describe each sentence.

Recommendation System

The recommendation system takes each sentence in the item space and its associated vector representation as input and uses heuristics to select a subset of sentences and recommend them to the user. We will explore several heuristics to make a recommendation. As shown on the right side of Fig. 1, the green rounded rectangle presents some possible recommendations using attributes we have. The first heuristics could be that the recommendations are made using only the attributes from the sentiment, emotion, or clustering as the dotted arrows indicate. For example, the system randomly selects one sentence from each type of emotions and then show six alternative next sentences to the user. In this way, the user could have various choices in story continuation with different emotional feelings and pick the one who thinks that fits best. Alternatively, we could use combinations of different attributes. For example, the system presents an angry and neutral sentence or a surprise but negative one. These are basic heuristics

Risk Name	Severity	Likelihood	Mitigation
Sentence generator not working as expected	High	Medium	Use alternative models
Pipeline implementation lasts more than expected	Medium	Low	Extend implementation phase by shortening evaluation phase
No sufficient participants in user study	Low	Low	Perform self evaluation

Table 1: Risk assessment of the project.

that we are going to explore. If completing the work in advance, we will further explore a smarter recommendation system that could learn from the user selections by incorporating a user preference model. There are two ways to collect user's information. One is to explicitly survey the user's preference before the story generation process. We could present sentences with different sentiments, emotions, genres, style or even different novelty degrees and ask them to pick the favourites. The survey might be annoying to some users as we repeatedly ask questions until we collect sufficient information to build the user model. Instead, the system could learn from the user's selection in the story generation process and update the recommendations in the next iteration based on what the user selects previously. To achieve this, we will use the techniques in recommendation system literature [24] to help build a user model and decide what sentences to recommend.

After implementing the main system processes, we will create a simple command-line **user interface** with python scripts for system interaction. Finally, we will also conduct a comprehensive evaluation of the system, consisting of **user study** and **automatic evaluation**, and **analyse the results**. The detailed plan is explained in Section 4.

3.3 Risk Assessment

As shown in Table 1, our project has three main potential risks with different degrees of severity and likelihood. Firstly, the generation of candidates sentences is essential in our system as it forms item space for the following recommendations. If the selected sentence generation model does not manage to generate different sentences for a given input or the generated results are poor, it will directly affect the quality of recommended sentences or even lead to a failure of the whole system. Therefore, the risk is of high severity. While we have identified several sentence generation model in the related section and four ways of generating different sentences. Thus, we can use alternative models with appropriate approaches to accomplish our generation task. Secondly, delays in the completion of pipeline implementation may occur since unexpected bugs and problems may arise at every stage of development and slow down the progress. To mitigate the risk, we could extend the implementation phase with the cost of the shortened evaluation phase. We could replace the time-consuming user study with an evaluation based on self-work examples with the system. A good analysis can also be made based on self-evaluation results. Lastly, there is a small risk that we might be unable to find sufficient participants to conduct a user study, especially during the pandemic. In case of the risk, we could take the same measure as in the second risk to perform an alternative self-evaluation.

3.4 Ethics

Our project contains a user study that involves human participants to evaluate the effectiveness of our system. Therefore, we will follow the School of Informatics ethics procedure and submit an Informatics Ethics Form for review to ensure that our study is compliant with the General Data Protection Regulation and the policy of the University of Edinburgh. For that, we will prepare a participant information sheet, consent form, and data management plan to seek participants' consent and ensure their rights and privacy are protected.

Besides, to minimize the probability of generating naughty words and sensitive speech that might harm our users, we will avoid using sentence generators pre-trained on datasets that might contain this kind of language or involve any political views.

4 Evaluation

We will perform automatic evaluation after each module of the main pipeline (e.g. text generator, sentiment classifier) is implemented for the sanity check. We will use the test set from the dataset the machine learning models pre-trained on (e.g. ROCStories [25]) to generate results for evaluation. Then, following related work [6][26][13], we will use three metrics (i.e. coherence, BLEU, and perplexity) for sentence quality, two metrics (precision and recall) for classification performance, and purity for clustering. For the evaluation of the whole system, as studied in See's work [27], the currently only reliable way to evaluate the text quality of the generated story is human evaluation. We will conduct a small user study involving ten human participants and provide them with our system with a command-line interface. We ask them to interact with the system to create stories and interview them about the experience. After collecting this data, we will analyze the stories example and participants' feedback to draw a conclusion.

5 Expected Outcomes

We expect the story generation system with continuation recommendation could be successfully implemented using the methods in the methodology section, which indicates our research addresses the formulated recommender system problem and we are able to explore the better approach for setting up the system. Furthermore, the results of automatic evaluation and user study are expected to be positive, implying the identified gap in story generation is well filled and thus creating the new paradigm in this research area for further investigation. The experimental results should also show further insights that set the stage for future work.

6 Research Plan, Milestones and Deliverables

The project can be divided into three large work packages: system implementation, evaluation, and dissertation. Since we plan to use pre-trained models to implement our system, there is no need for finding a dataset explicitly. We start with system implementation, consisting of four sub-tasks. The first subtask is developing our first system pipeline, which is the main pipeline of our system with one sentence generator, one evaluation technique, and one basic rule for recommendations discussed in methodology section. It takes a relatively long period (4 weeks) to complete since we build the system from scratch. In the second and third pipeline, more evaluation techniques and recommendations rules will be explored and added to the existing system, which is expected to be less time-consuming. After the full completion of system functionality, we will create a command-line based interface of the system for the subsequent user study by the middle of July.

The evaluation process starts from early July before completing the whole system. It includes user study, automatic evaluation, and results analysis. We will begin to find participants for the user study and distribute them participant information forms and content forms following the School ethics procedure. After the system completion in the middle of July, we conduct the user study and collect the results at the end of July. Concurrently, we will conduct experiments and evaluate the results using automatic metrics. The results analysis of both user study and automatic evaluation will be conducted at the end of July.

The third working package is the writing of the dissertation. We will start the writing from the beginning of the project and complete each section concurrently with its corresponding implementation process. The content of the introduction section will be updated periodically after completing each subtask. Thus, it lasts the whole period. We aim to complete the dissertation one week before the deadline and use the period for proofreading. In this way, we can ensure the writing quality and submit it on time.

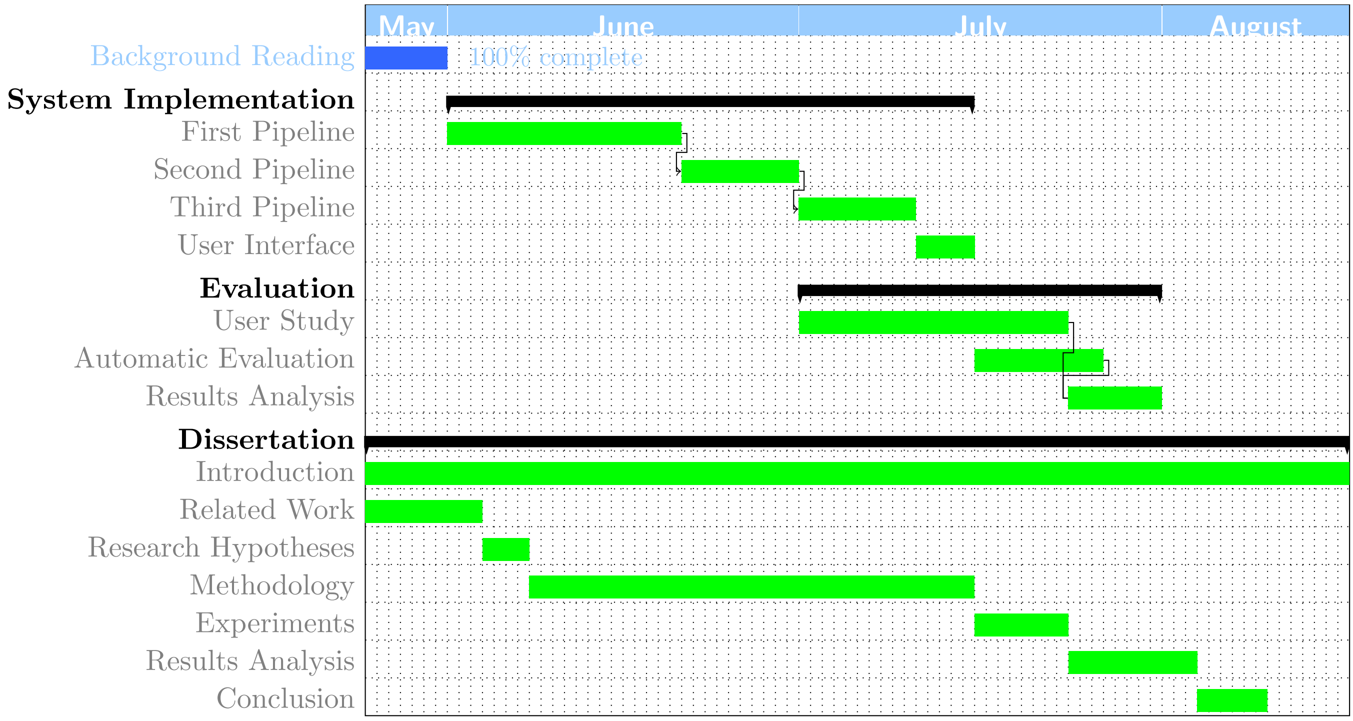

Figure 2: Gantt Chart of the activities defined for this project.

Milestone	Week	Description
M_1	6	System Implementation completed
M_2	7	User study completed
M_3	8	Evaluation completed
M_4	10	Submission of dissertation

Table 2: Milestones defined in this project.

Deliverable	Week	Description
D_1	3	First system pipeline
D_1	4	Second system pipeline
D_1	6	Third system pipeline
D_1	6	Command-line interface for user interaction
D_2	8	Evaluation report on user study and automatic evaluation
D_3	10	Dissertation

Table 3: List of deliverables defined in this project.

References

[1] Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. Better language models and their implications. *OpenAI Blog https://openai.com/blog/better-language-models*, 2019.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

[3] Mark Owen Riedl and Vadim Bulitko. Interactive narrative: An intelligent systems approach. *AI Magazine*, 34(1):67–67, 2013.

[4] Linbo Luo, Wentong Cai, Suiping Zhou, Michael Lees, and Haiyan Yin. A review of interactive narrative systems and technologies: a training perspective. *Simulation*, 91(2):126–147, 2015.

[5] Amal Alabdulkarim, Siyan Li, and Xiangyu Peng. Automatic story generation: Challenges and attempts. *arXiv preprint arXiv:2102.12634*, 2021.

[6] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385, 2019.

[7] Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. Plotmachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, 2020.

[8] Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J Martin, and Mark O Riedl. Story realization: Expanding plot events into sentences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7375–7382, 2020.

[9] Su Wang, Greg Durrett, and Katrin Erk. Narrative interpolation for generating and understanding stories. *arXiv preprint arXiv:2008.07466*, 2020.

[10] Baptiste Caramiaux Rebecca Fiebrink, Laetitia Sonami. Machine learning for musicians and artists. Kadenze. `https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists-v/info` Accessed April 4, 2020.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[12] Feifei Xu, Xinpeng Wang, Yunpu Ma, Volker Tresp, Yuyi Wang, Shanlin Zhou, and Haizhou Du. Controllable multi-character psychology-oriented story generation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1675–1684, 2020.

[13] Faeze Brahman, Alexandru Petrusca, and Snigdha Chaturvedi. Cue me in: Content-inducing approaches to interactive story generation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 588–597, 2020.

[14] Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A Smith. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, pages 329–340, 2018.

[15] Melissa Roemmele and Andrew S Gordon. Creative help: A story writing assistant. In *International Conference on Interactive Digital Storytelling*, pages 81–92. Springer, 2015.

[16] Folasade Olubusola Isinkaye, YO Folajimi, and Bolande Adefowoke Ojokoh. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3):261–273, 2015.

[17] Joseph A Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction*, 22(1):101–123, 2012.

[18] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

[19] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.

[20] John. Elements of a story. The art of narrative. `https://artofnarrative.com/2020/07/14/elements-of-a-story/` Accessed May 5, 2020.

[21] Klaus R Scherer. Emotion as a process: Function, origin and regulation, 1982.

[22] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[24] Paolo Viappiani and Craig Boutilier. Recommendation sets and choice queries: there is no exploration/exploitation tradeoff! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, 2011.

[25] Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. Lsdsem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, 2017.

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

[27] Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. Do massively pretrained language models make better storytellers? *arXiv e-prints*, pages arXiv–1909, 2019.