Character Depth and Sentence Diversification in Automated Narrative Generation

Brooke Bottoni,[1] Yasmine Moolenaar,[1] Anthony Hevia,[1] Thomas Anchor,[1] Kyle Benko,[1] Rainer Knauf,[2] Klaus Jantke,[3] Avelino Gonzalez,[1] Annie Wu[1]

[1]Computer Science Dept.; University of Central Florida; Orlando, FL, USA
bbottoni@knights.ucf.edu, ymmoolenaar@knights.ucf.edu, anthony.hevia@knights.ucf.edu,
thomas.anchor@knights.ucf.edu, kyleb3nko@knights.ucf.edu, Avelino.Gonzalez@ucf.edu, aswu@cs.ucf.edu
[2]Computer Science Faculty; Technische Universitat Ilmenau; Ilmenau, Germany
Rainer.Knauf@tu-ilmenau.de
[3]Adicom Group; Weimar, Germany
klaus.p.jantke@adicom-group.de

Abstract

This paper describes and discusses methods for improving character depth and sentence diversification in automated storytelling systems. The fAIble III system that is the subject of this paper addresses a major limitation of its immediate predecessor (fAIble II) in that the characters in its stories seemed to act in a vacuum, without any apparent reasons for their choices or emotions. This amelioration is accomplished through generating character backstories. fAIble III also addresses the diversity of generated sentences with a pattern recognition system that removes many of the awkward and repetitive sentences that drew negative comments in the testing of fAIble I and II. Lastly, stories generated by the three iterations of fAIble are compared and empirical test results are presented.

Introduction

Throughout history, people have told stories to pass down values, teach lessons, and inspire future generations. Tales both true and fictitious are woven into art, philosophy, and belief systems. There have been several attempts to automate the storytelling process, all of which achieved varying degrees of success. Some of these systems focused on high-level story planning, others focused on description of the story world and its elements, and still others focused on character development. A common thread throughout many of the recent systems is an abundance of repetitive sentence structures and a lack of reasoning behind character actions. Two of the main goals in the development of fAIble III were ameliorating these issues through the creation of a pattern recognition subsystem and a background generation component.

Background

One of the first attempts to automate the storytelling process was the Tale-Spin system in 1977 (Meehan 1977). This first system used a predefined set of rules and world qualities to generate various Aesop's fables. It was relatively narrow in scope, but it was an inciting force for the expansion of the field of automated narrative generation. As more and more research was performed on the automated generation of stories, systems began to specialize and focus their attention on one aspect of narratives. Systems such as UNIVERSE (Lebowitz 1985) and MEXICA-Impro (Pérez y Pérez and Lemaitre 2010) narrowed their scope to developing the characters in their stories, focusing on their emotions and goals. Meanwhile, systems such as MINSTREL (Turner 1991) and VB-POCL (Riedl 2010) focused on events, rather than characters, to drive the story. These systems often used traditional planning systems to determine the path of the narrative.

More recently, some architectures have employed variations of the classical models of AI planning. These include heuristic planning algorithms such as those found in CONAN (Breault, Ouellet, and Davies 2018), and the adoption of LSTM networks found in Plan, Write, and Revise (Goldfarb-Tarrant, Feng, and Peng 2019). CONAN uses a planning algorithm known as Fast Downward alongside the information it has about the story world and its characters' preferences to create a logical sequence of events that can result in an achievable quest. Porteous and Lindsay (2019) make the case that incorporating a process by which the antagonist continually seeks to interfere with the goals of the protagonist can reduce the burden on the human author (the user of the system). They view this issue as non-cooperative multi-agent planning.

However, there are a limited number of systems that generate backstories for their characters. Some narrative generation systems utilize emotions and goals for determining character actions, but these are not informed by a history of prior experiences, as is often the case in human-generated fiction, and indeed in real life. In analyzing most works of popular fiction, one will notice that almost all main characters have rich backstories that affect their personalities, goals, and actions. Such backstories are rarely part of automated narrative generation research. Therefore, our research presented here has sought to create a system that builds a scaffolding of experiences with which to reason about character-related plot points, and to provide more explanation for their decisions and actions.

History of the fAIble System

Our work is the culmination of a four-year NSF-supported research project. The AESOP storytelling framework (Wade and Gonzalez 2017) was the first iteration of the project, which was later named fAIble. This system manually creates a model of information that can inform the system when creating stories. The story world is limited to the nouns, verbs, and sentence structures implemented, and is governed by a set of rules. AESOP chose events semi-stochastically based primarily on individual event probabilities, each event's preconditions and postconditions, the performing character's attributes and goals, and previous events in the story. This system can only generate simple Subject - Verb - Object sentences, as seen in this story snippet:

> Harriette meets Gary. Gary becomes a friend of Harriette. Harriette becomes a friend of Gary. Gary antagonizes Harriette. Gary becomes an enemy of Harriette. Gary attacks Harriette. Gary attacks Harriette. Harriette attacks Gary.

The deficiencies of this system are evident from the story snippet above – repetitive, short, illogical, and sometimes contradictory events. Nevertheless, the work was valuable in exposing the advantages and limitations of using a probabilistic approach to event generation.

The fAIble I system (Kazakova and Gonzalez 2018) followed AESOP and was radically different. It focused on character-driven stories, goal-based decision-making, and was modular in its architecture. fAIble I continued to use probabilistic event generation, but incorporated a graph database for keeping track of context-based reasoning and character thoughts. The stories sounded less repetitive and richer in detail, and the language was more expressive as a result of an added Natural Language Generation (NLG) layer. An example of a fAIble I story snippet follows:

> Our story begins in a village. A mercenary named Nathaniel lived in the village. The village was welcoming. Nathaniel had a companion named Christopher. Daria, the creature, killed Christopher. Nathaniel set out to avenge Christopher. Nathaniel thought: "Is Nathaniel at location of Daria: forest?", "No" he thought. Nathaniel went to a forest. The forest was sunny. Nathaniel thought: "Is Daria dead?", "No" he thought. Nathaniel thought: "Is Nathaniel evil?", "No" he thought. Nathaniel thought: "Is Daria evil?", "Yes" he thought. Nathaniel thought: "Nathaniel has weapon?", "Yes" he thought. Nathaniel attacked Daria violently with a sword.

The stories generated by fAIble I are more diverse in their sentence structure and provide a more engaging plot than those of AESOP. Despite the awkwardness of the constant "internal thoughts" dialog, this feature was a step in the right direction for character depth and reasoning behind actions.

fAIble II (Alvarez and Gonzalez 2019) was a transformation of the fAIble system, drawing distinct lines between world, event, and sentence creation. Its modularity was increased, and fAIble II's event generation is less stochastic and more driven by the goals of the characters. A turn-taking system was implemented to alternate which character has agency to act. An event translation layer was added after event generation to filter the events and give them more structure before they're passed to the NLG module. The resulting stories focus on a quest and enhance the description of the world and how characters travel through it. Here is an example of a partial story generated by fAIble II:

> Our story begins with Q, a civilian. He cared for Kirk. There was also Khan, a sailor. He took Kirk. Khan looked for a way to cross an antica bay. He saw a motor boat. He boarded the motor boat. He escaped from Q into the antica bay. Khan infuriated Q. He was on a quest to get Kirk back from Khan. Q, a civilian, looked for a way to travel across the antica bay. He noticed a yacht. He boarded the yacht. He went from a shuttle port to the antica bay. He crashed his yacht on his way to a shipping yard.

While fAIble II succeeds in moving away from the distracting self-questioning and generally improves the quality of the stories by making them richer in detail, it still fails to provide sufficient information about the reasons for characters' actions.

Our modifications build directly upon the work of the AESOP (Wade and Gonzalez 2017), fAIble I (Kazakova and Gonzalez 2018), and fAIble II (Alvarez and Gonzalez 2019) systems. A main thread among these stories is that characters act based on a goal, but this is never explained to the reader. If the reader has no idea of the internal workings of one of these systems, an action by a character (such as kidnapping someone) may seem impulsive and illogical. One of our goals in the development of the fAIble III system was to create a structure that would allow characters' actions to be justified and reasoning to be presented in an eloquent way. In addition, the sentences of all three prior systems tend to sound repetitive, a complex issue that was addressed in a variety of ways in fAIble III.

System Description

The subject of this paper, the fAIble III system, is split into four main stages. Before a story is generated, a story world must be created to give the story a setting and the characters connections to elements of the world and each other. This stage includes the process of background generation, which is a new addition in fAIble III and allows for improved event reasoning. Next, events are generated based on a context-free grammar that allows for multiple sequential plot lines and subplots; these events are then passed to a translation module that converts the detailed event objects into sentence structures and uses a pattern-recognition algorithm to diversify the sentence composition. Finally, the sentence structures are joined into readable English sentences in the NLG module. In addition, a 3D avatar storyteller that reads the story text out loud through a text-to-speech system was added.

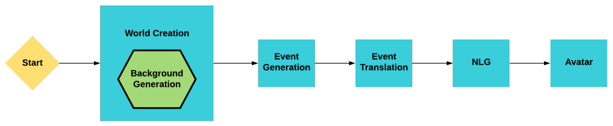

Figure 1: The Modules of fAIble

World Creation

The purpose of the first stage in fAIble III, world creation, is to decide what characters, assets, vehicles, and locations the story world contains and how all of the elements are connected. At the start of world creation, a token is generated and registered with the database, and a theme, such as "medieval" or "space opera" is selected at random. Based on the theme selected, the world creation component uses JSON-formatted schemas stored in S3 to construct the world. These schemas contain names and types of objects to ensure the story is internally consistent. Once a schema is selected, the component generates, in the following order: locations, vehicles, characters, and assets; it then initializes a quest by placing the protagonist into the world, and generates a background for the main characters (discussed in detail below). In each step, the component retrieves the appropriate schema objects from S3 and uses them to create and insert nodes in the graph database.

Event Generation

The next step of the fAIble III system is event generation, which pieces together the plot of the story by following an algorithm loosely modeled after the following replacement grammar:

```
S –> iPo
i –> intro
o –> outro
P –> eG | G
G –> fTR | TR | R
f –> search for information
T –> eT | PT | t
t –> travel
R –> r | rP
r –> resolve current plot
e –> good event | bad event
```

In this system, a story (S) consists of an intro, a plot (P), and an outro. A plot consists of some good or bad event happening to the protagonist (e) and the protagonist pursuing a goal (G). Pursuing a goal consists of gathering information (f), traveling (T), and a resolution (R). During travel, sub-events can occur, a subplot can be started, or the main character can simply travel to a location. During the resolution, the current plot is resolved, after which a new plot can be started.

A character's backstory and previous story events allow for the creation of "plot hooks" or the beginning of new subplots. For example, if the main character sets out on a quest to obtain a magical sword about which he or she had heard legends, a subplot where the antagonist steals the sword would be added to the possibilities for continuing the story. This plot hook structure not only allows for longer stories, but also makes story progression more logical. The compartmentalization of each life stage into a function and further separating life events into function calls allows for an easily debuggable and scalable system. Adding additional life events to the background structure is as simple as writing a new function and exporting it to the appropriate life stage.

Event Translation

The purpose of the event translation piece of the system is to convert the raw event details from event generation into JSON that can be fed into the NLG process. After the plot has been created by event generation, those events are passed to translation in the form of a JSON file. Then, translation breaks that file down into its individual events. In fAIble II, this was done by pulling out events one-by-one, processing them, and adding them to the output to eventually be passed to NLG. This often resulted in choppy sentences and made it hard to create more complex sentence structures. In fAIble III, this part of the system was restructured so the entire batch of events is processed together, and common sentence patterns are condensed into various structures. This pattern recognition process is discussed in more detail later. To process an event, the system uses the information in the JSON, which includes the type of event (e.g. "walk", "find", or "rescue") and the subject (e.g. "Maria" or "the sailor"), along with adverbs, adjectives, and other sentence parts. Based on the event type (or types, when combining sentences), this information is passed to a function that puts these sentence pieces in order and creates an object. NLG takes in the sentence objects from event translation and turns them into readable English sentences. This piece of the system was not changed from fAIble II to fAIble III.

Addressing Character Depth and Reasoning

Humans don't usually act without reason, and characters in a story shouldn't either if the story is to sound logical and internally consistent. The purpose of the background generation algorithm is to create a scaffold representing a character's past in order to provide logic and reasoning for that character's emotions and actions during the main story. We aimed to generate a "history" for our main characters before the story even began. Not only did this allow fAIble III to justify actions, it also became a way to lengthen stories by creating new instigating events and goals based on characters' personalities, emotions, and relationships.

The life of each major character is abstracted to a progression of discrete events split up into three stages: youth, teenager, and adult. Each stage contains a pool of possible events from which the system can select and add to a character's life. In addition, each stage has one or more events that must happen, and some events that are not allowed. The selection of events in background generation is semi-stochastic and very similar to the AESOP system. The pool of possible events is based on preconditions and postconditions assigned to an event; for example, in order to lose an item, a character must first have possession of it.

Once an event is selected from the set, graph operations are applied to mutate the world/character graph state so the event is now valid - this could be giving the character an item it did not have before, or turning a friend into an enemy. The result of the background generation process is a new, mutated "world" graph with more character connections to other elements of the story world and "reasons" explaining those connections. One important aspect of this process is that for each of the events that create relationships, whether with a friend, a family member, or an enemy, the character the connection is formed with is already a part of the story world; this way, these characters can show up later to influence the narrative.

The fAIble II system could generate sentences such as "Chris valued a sword" and "Marty stole his sword"; however, there was no explanation as to why Chris valued that particular sword, or why Marty would steal it. With the creation of the background generation system, those sentences would sound more like "Chris valued a sword because it was a gift from his father" and "Marty hated Chris because they got in a fight. He stole Chris's sword." These sentences still sound a little choppy, but they provide some information about the characters' motivations, creating a more logical storyline. We hope to build on this process in our future research to provide a richer description of the backgrounds of the main characters.

Addressing Lack of Sentence Variation

Our first step in reducing repetitiveness and increasing the quality of the sentences overall was analyzing how sentences are processed in event translation. Sentences in stories are affected by the sentences around them, and most readers can quickly notice if they don't flow together "correctly": e.g. "Noah went to the store. He went to the store to buy eggs." or "Sara saw a spaceship. She boarded the spaceship." There are quotation marks around "correctly" because prescriptively these sentences have correct grammar. However, they sound robotic and inhuman - "incorrect" in a rhetorical sense. The core issue with those sets of sentences is repetition. Most of the time, readers don't expect to hear the same noun-verb pair or direct object in two consecutive sentences. In reality, it is nearly impossible to correct these kinds of context issues without having any information about the surrounding sentences; therefore, the system needed to be modified in a way that would allow it to process multiple events at once.

For each of the action functions, we created another function to output a clause object instead of a whole sentence object; then, they could be used as building blocks for more complex sentences. The information contained in a clause object and a sentence object are nearly identical, but due to the structure of the system, multiple clause objects can be combined into one sentence, while sentence objects are standalone. Because the action functions were no longer returning entire sentences, mappers could be created inside actions that were the beginning of a common sequence, such as 'notice' then 'board'. Patterns would be recognized and combined into one or more sentences, while individual actions that were not part of a pattern were processed separately. Instead of outputting "Nico noticed the pirate ship. He boarded the pirate ship." the 'notice' - 'board' pattern would be combined into "Nico noticed and boarded the pirate ship." Similarly, the pattern 'notice' - 'get' would be condensed from "Ghita saw a sword. She picked up the sword." to "Ghita saw a sword and picked it up." All of the action objects passed to NLG were processed in this way, allowing for more fluent paragraph structures and decreased repetitiveness between sentences.

Additional smaller edits were made to increase the sentence diversity, including fixing the question and answer dialog. When a character is searching for something, they are able to ask other characters where it is located. There was a bug in fAIble II that caused side characters to never have the information the protagonist was looking for, so these sentences were never used. In fAIble III, we fixed this system to allow this dialog to happen sporadically, creating more interesting stories.

Comparison of Generated Stories

In order to demonstrate the changes made to the system, we implemented them on the intro section of fAIble II. This way, we were able to juxtapose intros generated by fAIble II and fAIble III, and gauge whether our modifications actually enhanced the system output. Below are examples of fAIble II intros and fAIble III intros, for comparison:

> fAIble II:
> Once upon a time there was John, a civilian. He valued a sword. There was also Khan, a sailor. He took John's sword.
>
> Our story begins with Q, a pilot. He cared for Kirk. There was also Thomas, a knight. He took Kirk.
>
> fAIble III:
> Once upon a time there was Martin, a prince. He valued a war hammer because it was a gift. There was also Anthony, a doctor. Martin disliked Anthony because they were lifelong rivals. Anthony took Martin's war hammer.
>
> Our story begins with Peter, a soldier. He cared for Kirk because they grew up together. There was also Alex, a knight. Peter hated Alex because they were friends but got in a fight. Alex kidnapped Kirk.

The syntactic complexity has increased in the last two intros. It still sounds a little telegraphic, but the additional subordinating clauses add to both the readability and the logic of the story. The further explanation of character actions leads to more character depth. An important part of stories is allusion, or indirect references to past events. An intro such as this one generated by the fAIble II system. . .

> Our story begins with Luke, a mercenary. He cherished an iron hammer. There was also Christine, a king. She took the iron hammer. She looked for a way to pass over a thar desert. She saw a carriage. She boarded the

carriage. She got away from Luke, a mercenary, into the thar desert. Christine infuriated Luke. He was on a quest to get the iron hammer back from Christine.

. . . would sound like this when generated by the fAIble III system:

Our story begins with Gwen, a blacksmith. She valued a rake because she went on a quest to find it. There was also George, a princess. Gwen disliked George because they were friends but got in a fight. George took Gwen's rake. George escaped from Gwen into a walkway. She decided to go on a quest to get it back from George.

While the fAIble II stories gave the reader all of the information about a situation up front, the fAIble III system is able to allude to a past quest and a fight between friends that was part of the characters' backstories. The background generation structure adds more complexity and internal logic to generated stories, as well as increased explanation of character reasoning, leading to deeper and more believable characters.

Testing and Results

Testing of the system was conducted through an anonymous online survey distributed to undergraduate students. Participants were presented with three stories generated by the fAIble system: one in print form, one in audio-only form, and one read aloud by the animated avatar shown in Fig. 3. 61 responses were recorded in total. The participants were asked eleven questions about the stories, nine of which were multiple choice, and two of which were free-response.

The first six multiple choice questions assessed the participants' perception of the quality of the stories, with the last three questions assessing the value of adding the avatar to the system. The first seven multiple-choice questions and the first free-response question were the same as those asked during fAIble II testing. The questions were as follows:

1. *Do story events appear to follow a coherent progression?*
2. *Do the characters appear to act based on some internal reasoning and motivations?*
3. *Do story events appear varied?*
4. *Does the language resemble human generated narrative?*
5. *Does the use of adverbs and adjectives add to the depth of descriptiveness of the story?*
6. *Is the language varied across sentences and stories?*
7. *Does the animation enhance the experience of the story?*
8. *Which do you like best? (Reading, Audio, or Audiovisual?)*
9. *Do you find the human avatar telling the story to be engaging?*
10. *What would you like to see the fAIble system do next?*
11. *How do you think the fAIble system could be improved?*

The participants answered each multiple-choice question with a 2 for "Yes", 1 for "Somewhat", and 0 for "No". The total score was the sum of all responses for a question and

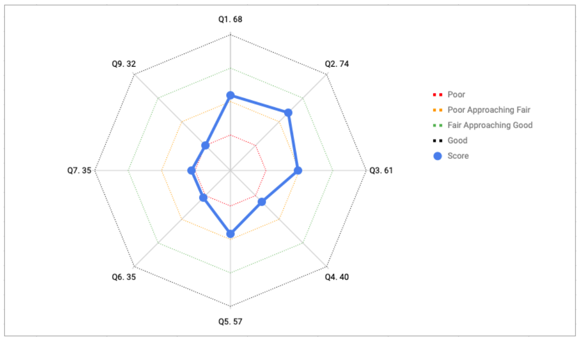

Figure 2: Survey Results

was broken down into Poor, Poor Approaching Fair, Fair Approaching Good, and Good correlating to the intervals: 0-32, 33-62, 63-92, 93-122. The results are shown in Fig. 2 below.

On the seven questions they had in common, fAIble III scored lower than fAIble II; however, there are several possible explanations for this. One of the goals of fAIble III was generating longer stories, and the stories presented in the fAIble III testing were around 20 sentences longer than the stories presented in the fAIble II testing. This could have increased the perceived repetitiveness of the fAIble III stories, which could account for some of the negative feedback related to event diversity and language variability. Additionally, we were not present when the research was introduced to the students completing the survey, since each survey was completed by a class of students and introduced by the instructor of that class. We have no knowledge about what was said to them before they began the testing, and the instructor could have potentially set a higher expectation by misstating the intent or scope of the research.

Many of the free-response answers focused on improving the avatar, adding more description and detail for environments and characters, and generating more complex sentences. Many other participants suggested improving the dialogue between characters and increasing the overall amount of dialogue. The questions that received the highest scores were 1, 2, and 3, which asked about story coherency, character believability, and event variability - three of our goals in the development of fAIble III. The testing results suggest that the fAIble system needs further development in order to generate stories that sound convincingly human. To generate longer stories, the pattern recognition piece of the system will have to be expanded and the quantity and variety of objects, people, and places the system has to choose from will have to be increased. Increasing the types of actions and sentence structures could also reduce perceived repetitiveness and improve the overall quality of the stories.

Future Research

There are two ways a system such as fAIble III can be enhanced in future research: with respect to its internal workings and with respect to the content from which the story generator pulls when creating a narrative. The former was

Q	fAIble I	fAIble II	fAIble III
1	1.39	1.00	1.11
2	1.27	1.48	1.21
3	0.98	1.12	1.00
4	0.29	0.79	0.66
5	0.88	0.93	0.93
6	0.46	0.75	0.57

Table 1: Comparison of testing results

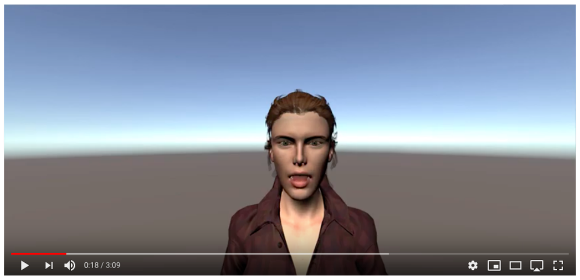

Figure 3: The Animated Avatar

the primary focus of this project; however, many of these modifications were done on a test basis and did not encompass the entire system. To further develop this system, the pattern recognition module should be expanded to recognize more, and increasingly complex, patterns. Background generation could be improved by generating backgrounds for characters in parallel, allowing characters' lives to affect each other earlier, thereby creating more complex relationships.

The reliance on authored content in a system like fAIble III is both a positive and a negative. While it's useful to be able to change the details of the generated stories whenever the user wants, that content must be written by someone, and it's required in order to generate completely new stories. The types of content that can supplement the system are numerous and include characters, their backgrounds, the settings in which they act, the items with which they can interact, and the adjectives used to describe everything. These all require human effort to create. An interesting idea for future expansions would be to create a webpage where users could upload their own story world schemas and generate stories whenever they want.

Conclusion

The fAIble III system generates logical, grammatically correct, often interesting, relatively detailed stories, the majority of which contain believable locations and items, as well as characters who seem to have motivation behind their actions. It improves upon past systems by generating a backstory for main characters and alluding to these past events, creating character depth and improving logical flow. Additionally, a pattern recognition component was added that increases the variation of sentence structures and make generated stories more interesting to read.

Acknowledgments

This material is based upon work sup- ported by the National Science Foundation (NSF) under Grant No. 1458272. The presented research is a contribution to the NSF Program International Research Experiences for Students (IRES).

References

Alvarez, M.J., A. R. B.-K. M. J. K. R. J. K., and Gonzalez, A. 2019. Hello, narratives: Character development in automated narrative generation. In *Proceedings of the 32nd Annual Florida Artificial Intelligence Research Society Conference*.

Breault, V.; Ouellet, S.; and Davies, J. 2018. Let conan tell you a story: Procedural quest generation.

Goldfarb-Tarrant, S.; Feng, H.; and Peng, N. 2019. Plan, write, and revise: an interactive system for open-domain story generation.

Kazakova, V.A., H. L. P.-A. G. L. K. R. J. K., and Gonzalez, A. 2018. Let us tell you a faible: Content generation through graph-based cognition. In *Proceedings of the 31st Annual Florida Artificial Intelligence Research Society Conference*.

Lebowitz, M. 1985. Storytelling as planning and learning. *Poetics* 483–502.

Meehan, J. 1977. Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, 91–98.

Porteous, J., and Lindsay, A. 2019. Protagonist vs antagonist provant: Narrative generation as counter planning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, 1069–1077. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Pérez y Pérez, R., N.-Y. S. E. P. R. C. V., and Lemaitre, C. 2010. Mexica-impro: A computational model for narrative improvisation. In *Proceedings of the International Conference on Computational Creativity*, volume ICCC-10, 90–99.

Riedl, M. 2010. Story planning: Creativity through exploration, retrieval, and analogical transformation. *Minds and Machines* 20:589–614.

Turner, S. 1991. A case-based model of creativity. In *Annual Conference of the Cognitive Science Society*, volume 13, 933–937. Association for Computing Machinery.

Wade, J., W. J. W.-M. P. L. J. K. K. R., and Gonzalez, A. 2017. A stochastic approach to character growth in automated narrative generation. In *Proceedings of the 30th Annual Florida Artificial Intelligence Research Society Conference*.