The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

Hello, Narratives: Character Development in Automated Narrative Generation

M.J. Alvarez,[1] R.E. Amaya,[1] K. A. Benko,[1] J. T. Martin,[1]
R. Knauf,[2] K. P. Jantke,[3] A. J. Gonzalez,[1]
[1]Computer Science Dept.; University of Central Florida; Orlando, FL, USA
[2]Computer Science Faculty; Technische Universitat Ilmenau; Ilmenau, Germany
[3]Adicom Group; Weimar, Germany
matt@mattjalvarez.com, Rebeca_amaya@knights.ucf.edu, kyleb3nko@gmail.com,
rainer.knauf@tu-ilmenau.de, klaus.p.jantke@adicom-group.de, Avelino.Gonzalez@ucf.edu

Abstract

Development of interesting and complex characters is the most important element of a narrative. Presented in this work is fAIble II, an automated narrative generation system that focuses on character development. fAIble II leverages a graph database, containerized modules, knowledge templates, and language structuring to produce diverse and coherent stories. Story progression is driven by character perception, emotion, personality, and interaction with the story world. The resultant system has been tested via anonymous questionnaire. Responses suggest its ability to create diverse, sensible narratives using character development.

Introduction

Time–hours, days, years. We never seem to have enough. Yet we spend so much of it for the sake of hearing, watching, or reading stories. People do this because they are looking for meaning, an escape, they are looking for a connection. They are connecting to people, people they can relate to, people they want to be like, deep characters who reflect a reader's desired self. As the character develops, the audience's relation with the character grows as well. It is that character development, the growth, that drives people to love stories. It is also that connection that can be seen universally in many great literary works. The fAIble II system strives to automate stories that elicit those relations with its characters. But this goal and project did not begin with us.

The automation of the creative process for story generation has been a topic of interest since the late 1970s. The goal of automated narrative generation systems is to create novel stories that are coherent and can maintain reader interest. Systems such as TALE-SPIN (Meehan 1977) pioneered research in this field and began the process of breaking down storytelling into computerized tasks. TALE-SPIN's main limitation was its dependence on using pre-defined rules that restricted the system's expansion. However, since then a significant amount of work has gone into automated narrative generation. While there are many notable works that have contributed to this field, fAIble II has based itself on the research of CAMPFIRE (Hollister 2016), AESOP (Wade et al. 2017) and fAIble I (Kazakova 2018) in particular.

The CAMPFIRE system focused on dynamic story generation that allowed users to change story details in real time. CAMPFIRE generates bedtime children stories using a Cooperating Context Model to create stories without the use of pre-determined paths. It was successful in generating stories with interesting quest-based plots and narrative language, but a noticeable pattern can be discerned when comparing multiple output stories. The system also relies on a large base of pre-authored material that makes expanding the story types time consuming.

The AESOP storytelling framework takes a different approach and generates stories using a stochastic process with probabilistic logic. A story space model is used to choose the events of the story and guide the narrative generation process. Character attributes are used to mold stories that are unique to each character and their story world perspective. This results in novel stories that do not rely on substantive amounts of pre-authored material. However, the stories tend to be short and lack consistency.

fAIble I used graph based reasoning to logically develop a narrative. An acting-thinking graph coupled with context-based reasoning and templates of common sense were used to generate a logical progression of events. While actions in the story make logical sense and characters act in their own self interest, the resulting stories needed continued development and natural language processing. In this paper, we present fAIble II, the next iteration of the fAIble I system.

CAMPFIRE focused on narrative progression and plot development, yet lacked dynamic character development(i.e. different character relationships, personalities, and emotions from one story to the next). Both AESOP and fAIble began the work towards this goal. AESOP used probabilistic story world logic and character attributes while fAIble I used an acting-thinking graph to abstract general story situations to generate character actions.

These systems set the foundation for character relationships and goal driven actions. fAIble II builds upon this by adding character perception and emotion as the motivating factor behind character actions. To achieve this level of character development, fAIble II moves the state of the story

world into a graph database, in contrast to fAIble I which used the graph database for decision making. This allows us to represent cascading character relationships, not only with other characters but also with other story components. As story events occur, these relationships evolve and influence future character decisions.

fAIble II is a part of ongoing research to develop a narrative generation system for children's bedtime stories. fAIble II does not base itself on previous models, but rather endeavors to explore the automation of story development and the creative process. The system is comprised of five isolated modules: World Creation, Event Generation, Event Translation, Natural Language Generation, and Animation. This paper contains discussions of each component of the fAIble II narrative generation system, and presents system testing and results.

Knowledge Base and World Creation

In order to create a mechanism in which we could reason about character relationships and emotions, we first had to represent the world in which the characters exist. Our team broke down the components of what a character could interact with into three categories: Locations, Characters, and Assets. The abstraction of all story elements with which characters can interact was necessary to properly represent character relationships with the story world.

Before the process of developing the narrative begins, the fAIble II system first creates a detailed world in which the characters will exist; World Creation begins with creating Locations, Characters, and Assets. The classification of all story entities into these three categories was important because we could now create rules (or implications) that belong to these story entities. For example, locations have properties of positioning in relation to other locations, and Characters have the property of being located at certain locations.

Locations Representing where assets and characters can exist in a story initially began as a flat graph in which locations were nodes and edges were connections between the locations. This flat representation was effective for reasoning about where characters can travel from their current position; however, this approach limited the richness the fAIble II system was seeking to achieve.

To achieve the level of detail required to describe locations, locations need to be represented on multiple levels. Location structures with multiple levels allow fAIble II to describe where characters are specifically located and in what area they are located. As an example, a character can be located at a castle in the story world; however we also want to be able to describe where that castle itself is located, such as in a kingdom or township. The first step in creating this type of location detail was to separate locations into two classifications: Abstract Locations and Concrete Locations. An Abstract Location is any location in which a character can exist within but not be located at (such as a city or country). A Concrete Location is any location a character or asset can be physically located at (such as a castle or building).

The next step was to define ways in which locations are connected and implications behind the transition between locations. Transitions were defined to be nodes that connect one concrete location to another concrete location. Since entities in the story world can only be located at concrete locations, the only way in which entities can move becomes via some type of transition. However, randomly assigning transitions to connect locations would create worlds that simply do not make sense. As an example, two rooms in the same building could be connected via an ocean. To solve this problem we created a few implications of traveling and connecting locations.

- Concrete Locations are connected by transitions
- Transitions have modes of movement associated with them
- Entities are restricted to a finite set of modes (i.e. a boat can sail across an ocean but cannot drive on an expressway)

With the creation of modes and transitions, we now need to update our definition of a location to include modes in which you can enter and leave a location and modes that are acceptable inside a location and all of the locations contained within.

Characters Characters can have roles, titles, gender (for sentence generation) and basic health statistics. Upon character creation, a sub-graph called the Mind is created and attached to the character node. The Mind Graph is the driving force behind character development, perception, emotion, personality, and eventual story progression. It is composed of nodes that represent any given character's understanding of entities in the story world. The Mind Graph can be considered the collection of a given character's understanding of their world. This graph only contains locations, assets, and characters that a character has seen before; this mechanism gives fAIble II the ability to tell what knowledge a character has about the world around it.

The Mind Graph is generated by a set of actions that are performed on a given character. The action inserts all nodes within one edge length of the character's location into the mind of the character. If a node already exists in the Mind Graph, the properties of that node will be updated. Upon creation of any nodes in the graph, new emotions are then connected to the character based on the character's personality. Emotions and personality are the driving force behind character development; they allow the fAIble II system to make decisions based on a character's experiences rather than solely on a character's actions and intentions.

Assets Assets are considered to be anything in the story world that are not locations or characters. Much like the modes described for locations, assets have been restricted to a finite set of generic classifications which can represent a wide variety of things that impact the narrative development. The classifications of assets are listed below:

- Weapons - Can be used to harm other entities in the story world
- Tools - Can be used to heal objects in the story world
- Medicine - Can be used to heal characters in the story world
- Vehicles - Give characters the ability to travel in a specific mode of transportation. (i.e. a sailboat allows a character to sail across an ocean)
- Keys - Can be used to open locked locations

Event Generation

Event Generation takes the graph structure created by the World Creation process and performs operations on the graph to further the narrative. The event generation system utilizes three main ideas to accomplish this: Agency, Character Goals, and Character Perception. These give the fAIble II system the who, what, and why of every action taken in a narrative.

The fAIble II system goes through three main phases during story generation: Introduction, Plot Development, and Conclusion. These three phases all utilize Agency, Character Goals, and Character Perception to generate events; however, the mechanism behind each phase is slightly different to properly begin and end the narrative.

Agency Only one character is given agency at any given time during event generation. This character is called the Agent or Active Character. Since fAIble II was designed with character interaction and development in mind, the system needed the ability for all characters to act autonomously in the story. All stories begin with a main character which is the first character granted agency. Agency can then be passed from one character to the next based on the active character's interactions with other characters. One example of passing agency is when a question is asked. When one character asks another character a question, agency is passed to the receiving character. The receiving character is marked as the agent and the system will generate events based on that character's personality and emotions which have been influenced by the character's previous interactions in the story. Agency is then passed back to the asking character once the receiving character has finished answering the question that initiated the transfer of agency.

Goals Understanding how fAIble II generates events requires understanding the specifics of the goal system used. For lack of a better term, *goal* was used to describe the current idea the author of the story is trying to convey. Goals are stored in the graph via a character's mind; The current agent's Mind Graph will have goals in the form of a linked list. The event generation process queries the "active" character and processes the most recently appended goal of the list. If the active character has no goals left to process, agency will be passed to the main character of the story. Every goal contains implications that can generate a variety of events. Furthermore, some goals have implications from previous events. It is also important to note that when events are generated, all of the subjects of the event and emotions the agent associates with the event, its subjects, and concepts are also included.

During goal processing, events are generated and new goals might be created. As an example, if a character encounters a locked door, a new goal created for that character might be to find the door's key. This goal system allows fAIble II to create logical and sensible events that further the narrative.

Character Perception As described in the world creation section, the Mind Graph is the driving force behind character development. Since every character has a Mind Graph, every character also has a concept of the story world structure. The Mind Graph also contains a few more nodes that represent more abstract concepts that drive character actions, development, and progression.

The first of the abstract components is the personality node. Every Mind Graph has one personality node that contains six basic emotion levels: Joy, Sadness, Anticipation, Anger, Fear, Disgust, and Trust. The character's predisposition to each emotion are noted zero to fifty.

The next abstract concept is emotions. Every element in the Mind Graph (Characters, Locations, Assets, and Goals) contains an edge that connects it to the character to which the mind is associated. These edges represent the character's emotions towards the element (e.g. a Character's emotions towards another character). The emotion edges contain the six basic emotions, however, the values of each emotion range from zero to one hundred rather than zero to fifty. These emotions are calculated by taking the character's respective personality value and adding a random number between 0 to 50 to it. This makes it so every emotion is influenced by that character's personality.

The last abstract concept is the concept node. The concept node works similarly to the emotion edges. The only difference is that the concept node represents non-material entities in the story world. Concept nodes are considered to be a character's emotions toward abstract ideas or actions (like a character's fear of flying). As such, concept nodes are calculated using the same mechanism as the character emotions.

When fAIble II enters event generation, character personality and emotions are always taken into consideration. Should some severe action happen, a character can change emotions towards entities or concepts in the story world. As an example, if a character is flying in a plane and the plane crashes, the character will then experience heightened fear of flying. Furthermore, if a character is afraid of flying, they might choose to sail a boat to cross an ocean rather than fly a plane. The constant updating of emotions combined with the use of emotions for decision making is what gives fAIble II the ability to produce character development and portray character emotions.

Introduction and Conclusion

The introduction phase serves as a mechanism to ensure the proper story state for plot development. The conclusion phase takes the output of plot development and creates a proper ending for the story. These phases simply query the world state looking for key observations, and generate events from a set of acceptable events to take place for a story. The introduction and conclusion sections provide brief story sequences and are a good subject for further study.

Plot Development

Plot development is comprised of a set of possible goals and events happening to characters in the story. It is important to note that there is no set ordering of events that must happen; however, there is a partial ordering of goals that must be processed after some event happens. Since the Introduction phase ensures a proper story state and beginning set of events, plot development performs iterations of goal processing on the active character in the story. The primary function of

plot development is to create a logical set of events for the narrative and to create a story that is not repetitive.

Interrupts Interrupts are a mechanism that breaks up sequences of repetitive events in the plot development phase. Every time a goal is processed, the plot development process tracks how many consecutive attempts have been made to resolve a goal without interruption (i.e. a character keeps on traveling from one location to the next without any other events happening). After several unsuccessful attempts to resolve the same goal, the system will create an interrupt.

An interrupt is a piece of knowledge that is supplied by the knowledge base. Each interrupt has certain implications associated with it. The implications associated with interrupts are a sequence of actions to be performed on the story world (e.g. a storm interrupt requires the world has a storm happening at some location). After an interrupt has been created and the implications of the interrupt have been applied to the world graph, a goal is appended to the active character's goals. This added goal represents a new situation that must be addressed by the character. As an example, if a character is sailing across an ocean, a whirlpool might be an interrupt created; the character is then required to pass by the whirlpool or overcome the obstacle in some manner.

Event Translation

Writers can write. Computers however, need to overcome the linguistic knowledge gap. Event Translation is where the events start their process to becoming readable text. To aid in explaining the process, look at the following sentence: "Alice the warrior princess has entered a red dragon's lair, sword in hand, scared but ready to fight." The Plot Development section noted that sentences like the one just described are not what Event Generation outputs; rather, the output is an object containing every detail to that sentence. In that object could be all the places Alice could get to from the lair, the details of the dragon (color, weight, abilities like breathing fire), all of Alice's emotions related to the dragon (her disgust and fear), and how skilled she is at sword fighting. That data needs to be parsed for the most relevant information and reformatted for The Natural Language Generation module.

Given an event that contains all possible information, a decision has to be made on how to convey the event to the audience. The Event Translation module does this by taking on the role of a story author, parsing events by type so that the sentence structure for the event can be written. fAIble II decouples Event Generation and Natural Language Generation, making Event Translation essential to the transition between the two components. There are many grammatical and structural nuances that can occur in any one event. Attempting to pass a generated event straight to the Natural Language Generation module can lead to errors in sentence construction as well as delays for new event types.

Framing Translation begins with a framing process, where sentences can be added to the beginning and/or end of an event. Using Alice again, we could explain the feelings she has on her approach to the cave, followed by a physical description of the cave. Those descriptions frame the actual event and improve narrative clarity. Other systems used framing techniques, such as CAMPFIRE; however, authorship of the before and/or after was for the whole story template. In fAIble II, because framing is contained in only one event, a programmer can author just one framing and have it work for all future stories. That significantly reduces the time it takes to expand variability, as well as allowing a more rapid elimination of bad sentence variety, by consolidating errors to a scene verses an entire narrative. With the appropriate framing in place, the system can start refining how a sentence is phrased.

Phrasing A talented author can make you cry with a character. The writer achieves that reaction by phrasing the emotion so that the audience perceives the emotions along side the character. Look at the following two sentences, "Alice's father was killed by the dragon causing her to melt to the floor with grief." and "Alice's father was killed by the dragon causing her to feel sad." Emotional delivery matters, grief is a very precise sadness. Authors can deliberate for hours on the right phrasing to get the perfect emotion across; fAIble II has begun the work to abstract this process into an algorithm. To begin, characters have personality (a set of base emotions) described in Character Perception in Event Generation, allowing fAIble II to replace emotional words by querying an emotional database. The query checks for an appropriate emotion word based on the word's part of speech and emotional intensity of the character. This lets fAIble II show emotional development in the character at a surface level. Once the best emotional word is injected into the output structure, minor phrasing adjustments are added based on the audience's context of the story (who and what the character has seen, where the character has been) to make the final sentence output resemble natural language.

Generic Sentence Structure Once the phrasing details for a particular event have been decided, they need to be constructed appropriately so that the event can be converted to English text. To do this, a generic sentence structure (GSS) constructor was created. The GSS constructor can make sentence structures for four categories: simple, compound, complex, and compound-complex. fAIble II's natural language generation library, discussed in detail in the next section of the paper, can generate simple and compound sentences (one or more simple sentences). Simple sentences are broken down into basic building blocks: subjects, verbs, direct objects, indirect objects, and complements. The GSS constructor follows basic English grammar rules to allow for virtually any sentence to be created. The restructured event is then passed to the natural language generation component to be processed into the final text output.

Natural Language Generation

The natural language generation (NLG) of a narrative system plays a key role in the delivery of the stories generated. Having natural sounding sentences that properly convey meaning is of vital importance to ensuring that the whole story comes together. This can be achieved by having pre-written text that replaces events. However, this solution increases the authorial burden of the system and is not ideal. Dynamic language

generation is preferred because it gives the system flexibility and does not limit the sentences to a specific event, genre or use case.

Consequently, the goal of the NLG component of fAIble II was to allow for dynamic sentence creation that did not restrict the types of sentences that could be generated. To meet this goal, fAIble II created a library of functions that are used to compliment the simpleNLG API (Gatt & Reiter 2009). simpleNLG is a realization engine that generates English text using a syntax-based format. It uses a lexicon and morphology system to conjugate and apply general grammar rules. fAIble II's auxiliary NLG library was needed because simpleNLG must be provided the specific syntactic parts of a sentence to generate the corresponding text. Since the English language is extremely diverse, creating the components for every possible sentence is cumbersome and implausible. Rather than having to manually input the sentence parts into simpleNLG, the NLG library uses a generic sentence structure provided by the Event Translation's GSS constructor and then interfaces with simpleNLG to convert to text. This generic sentence structure exploits the underlying pattern of sentences, allowing grammatical structure to be extracted for dynamic sentence creation that is not limited by the system's purpose. Thus, given the correct pieces, fAIble II's new NLG library can generate sentences easily and intuitively.

The NLG library could ultimately be used as a language generation tool in other systems not related to narrative generation because it is not use-case specific. Given a generic sentence structure input, virtually any sentence can be generated. However, simpleNLG continues to be an important component because it provides the logic behind syntax, conjugations, and punctuation of sentences.

Other narrative generation systems have used simpleNLG, including the original fAIble system. The previous fAIble system's NLG component focused on redundancy reduction, character identifier variation, and descriptor-based enhancements. These features are now handled by the Event Translation component, allowing the NLG component to focus purely on generic sentence construction with structural modifications for sentence variation.

Sentence Variation fAIble II's NLG library can generate simple and compound sentences that provide variety in the narrative. However, within these sentence types, more variation can be added by changing the sentence structure. Two additional sentence structures were added based on the original simple sentence: sentence with front modifier and sentence with end modifier.

Sentences with front modifiers move the adverb to the beginning of the sentence and sentences with end modifiers move the adverb to the end of the sentence. For example, the sentence "the wind suddenly rushed through the open window" can be turned into "suddenly, the wind rushed through the open window" or "the wind rushed through the open window suddenly". The sentence structure is randomly assigned during the sentence realization process.

Other sentence structures, like inverted sentences and sentences with a front preposition, were explored; however, they lead to sometimes generating incoherent and mechanical sounding sentences that detracted from the narrative experience when assigned at random. In future iterations, more research could be done to decide when these sentence structures can be used.

Animation

As an auxiliary component, a visual system was developed to present the text of these generated stories in a more natural form. The original raw text can be read and understood by users, but the intention of the visual is to make the content more digestible and perhaps enhance the reader's grasp on the story itself. The visual component was designed to be a three dimensional model of a book in which the text of the story is printed on the pages. The pages can be flipped through with the press of a button, simulating the reading process. Audio voice-over of the story can be heard with each page to aid with reading. The visual and audio combination is the end product that is then given to the user.

When users interact with the visual of a 3D book, it is expected that they will become more engaged as it is much more pleasant to read from this visual than raw text. Furthermore, the visual will allow for the story to be chunked into pages and even further into smaller paragraphs. The animation aims to address the topics of enjoyment, focus, and a sense of physicality.

System Testing & Results

Testing was conducted through an anonymous online survey that was distributed to a diverse population of working class and undergraduate students. Seventy-five subject responses were gathered in total.

The system was evaluated for quality of event generation, natural language generation and the additional animation component. Participants were shown two randomly generated stories from the fAIble II system with different themes. The participants were also shown a video of the storybook animation of one of the stories. They were then asked eight questions regarding their opinions on the material shown. Questions 1–3 targeted event generation, questions 4–6 targeted natural language generation, and question 7 targeted the animation component. These 7 questions were multiple choice with possible "Yes", "Somewhat", or "No" responses. The eighth question was open-ended.

The first seven questions were scored out of 150 possible points using the grading scale: 2 for "Yes", 1 for "Somewhat" and 0 for "No". A question could score all 150 possible points if all 75 participants answered "Yes". The total score was then broken down into Poor, Poor Approaching Fair, Fair Approaching Good, and Good ratings correlating to the intervals: 0-37.5, 37.6-75, 75.1-112.5, 112.6-150. Figure 1 displays the questions and results for each with respect to this scoring scheme. No question received a Poor or Good rating. Questions 1, 4, 5, 6 and 7 were rated as Poor Approaching Fair. Questions 2 and 3 were rated as Fair Approaching Good.

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. *What would you like to see our system do next?*

The eighth, open-ended question asked what participants thought the system should work on next. For event generation, participants indicated that the current plots are too simple and repetitive. For the language component of the stories, most participants stated that they would like to see longer and more complicated sentence structures. They also noted that many sentences started with "he" or "she" and caused confusion about which character was being discussed. The participants also perceived the animation component as too simple and recommended including pictures, character animations, and voice audio.

Overall, the responses for the survey questions were mixed, with most falling under the Poor Approaching Fair rating. The event generation questions rated higher than the language and animation component questions. In general, event generation was rated as Fair Approaching Good while language and animation were rated as Poor Approaching Fair. The responses from the eight questions back up the ratings received for each component. Working on increasing the quality of the language generation and expanding the animation component will be main topics of interest for future iterations as they scored the lowest.

Similar testing was conducted for the fAIble I system. The same first six questions were used in an anonymous survey that was distributed online in a post about automated story generation. Thirteen responses were gathered in total. This sample size does not allow us to make any statistically significant conclusions about the first system. Nevertheless, similar to fAIble II, the story event generation scored better than the language generation. Due to the limited sample size, it is difficult to measure the amount of progress between the first and second fAIble systems. However, based on the testing results from fAIble II, we can conclude that the system performs to par with fAIble I and a solid foundation has been established for future progress.

Conclusion

fAIble II is still in development, however, the ground work is laid for future enhancements to be performed. fAIble II is completely decoupled and asynchronous in its execution. The separation of the components of fAIble II makes it highly extendable and scalable. While fAIble II features coherent character actions, logical progression of events, and character development, there is still much work to be done to produce high quality stories.

Acknowledgments This material is based upon work supported 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).

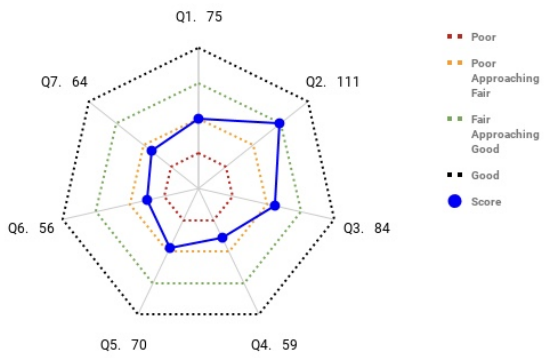

Figure 1: Question Score by Rating

References

Gatt, A., and Reiter, E. 2009. SimpleNLG: A realisation engine for practical applications. In Proc. 12th European Workshop on Natural Language Generation, 90–93. Stroudsburg, PA, USA: ACL.

Hollister, J. 2016. A Contextual Approach to Real Time, Interactive Narrative Generation. Ph.D. diss., Dept. of Computer Science, University of Central Florida.

Kazakova, V. A., Hastings, L., Posadas, A., Gonzalez, L. C., Knauf, R., Jantke, K. P., & Gonzalez, A. J. 2018. Let Us Tell You a fAIble: Content Generation through Graph-Based Cognition. Proceedings of the Thirty-First International Florida Artificial Intelligence Research Society Conference, 282–287. Melbourne, Florida: FLAIRS.

Meehan, J. 1977. TALE-SPIN, An Interactive Program that Writes Stories. Proceedings of the 5th International Joint Conference on Artificial Intelligence, 91–98. Cambridge, Massachusetts: IJCAI.

Wade, J., Wong, J., Waldor, M., Pasqualin, L., Jantke, K., Knauf,R., and Gonzalez, A. 2017. A stochastic approach to character growth in automated narrative generation. Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, 152–157. Marco Island, Florida: FLAIRS.