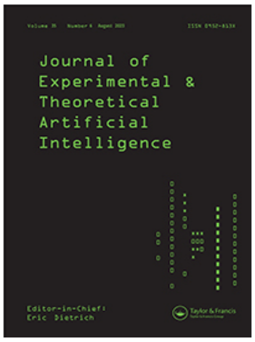

Journal of Experimental & Theoretical Artificial Intelligence

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/teta20

State-of-the-Art in Automated Story Generation Systems Research

Rebeca Amaya Ansag & Avelino J. Gonzalez

JOURNAL OF EXPERIMENTAL & THEORETICAL ARTIFICIAL INTELLIGENCE
2023, VOL. 35, NO. 6, 877–931
https://doi.org/10.1080/0952813X.2021.1971777

Taylor & Francis
Taylor & Francis Group

ARTICLE

State-of-the-Art in Automated Story Generation Systems Research

Rebeca Amaya Ansag and Avelino J. Gonzalez

Department of Computer Science, University of Central Florida, Orlando, FL, USA

ABSTRACT

This paper presents a review of research works from the last several years in automated story generation systems. These systems are categorised into interactive story generation systems and non-Interactive story generation systems. Interactive systems are those that collaborate with a user/author during the process of creating and/or executing the story. The extent of user interaction varies across systems but remains an integral part of the creation and/or the unfolding of the story. Non-Interactive systems concentrate on complete automation of the creative process involved in narrative generation to create diverse and interesting stories. Interactive story generators specifically designed for video game narratives are reviewed as a separate sub-class of interactive story generation systems. Also reviewed are the methods used for evaluation of story generation systems as a way to explore the possibility of having standard methods of evaluation within the research community. The paper includes a discussion of trends and directions of the research discipline.

ARTICLE HISTORY

Received 29 April 2020
Accepted 21 July 2021

KEYWORDS

Automated narrative generation; interactive narratives; automated story generation; storytelling systems; *computational creativity*

Introduction

Telling and listening to stories have always been a fundamental part of the human experience. Throughout history, storytelling has evolved with us, from an oral tradition to written manuscripts, live theatre, music, cinema, and other artistic representations. Stories have been used as a means of passing on history, culture and morals in an engaging medium. It also allows us to connect with others and learn about the world.

Stories in different forms also serve to entertain us – we all love a good story. Stories have traditionally been in the form of novels, movies, plays, comic books, and songs. They are and have been an important element of the entertainment industry. More recently, advances in computing have brought the art of storytelling to computer games, opening yet another avenue for storytelling in entertainment.

Moreover, as we learn more about the way the human brain functions, applications of storytelling have expanded into education and training. In neuroscience studies, researchers have found that reading words activates the sensory areas of the brain associated with them, as if the reader was truly participating in the narrative (Boulenger et al., 2008; Paul, 2012). Therefore, storytelling engages the brain on a deeper level than just linguistic processing. The readers can immerse themselves in the story and relate it to themselves. This can create a meaningful experience with readers who might be making connections with the story at several different levels. These connections help form a rich learning environment that is not only effective but also enjoyable (Caine & Caine, 1991).

CONTACT Avelino J. Gonzalez avelino.gonzalez@knights.ucf.edu Department of Computer Science, University of Central Florida, Orlando, Fl USA

Furthermore, stories can be used to teach a concept for better recall, as shown in a study by Bower and Clark (1969). Therefore, enhancing education with storytelling is indeed a worthwhile goal, and as a result, storytelling has become an increasingly popular educational vehicle for both children and adults.

Training people through their participation in simulated events has been a staple of professional training for many years in several disciplines, including military tactical training, airline pilot training, and for training first responders in the management of response to major emergency events. These simulations typically require a background story to accompany the training exercise in order to provide context and enhance its realism. These stories can be as simple as flying an airplane from airport A to airport B through inclement weather, or as complex as providing the background for the setting of a natural disaster, such as a major hurricane or earthquake.

As storytelling becomes an increasingly used tool in our modern society, a bottleneck occurs in creating sufficient stories for their different purposes. It has been argued that the consumption of stories, both for pleasure and for serious purposes, far outnumbers what is currently produced by human authors (Riedl, 2010). Creating computer programs that can automatically compose stories (we refer to this here as *automated story – or narrative – generation systems*, or more succinctly, *story/narrative generation systems*) could address this problem; it could also allow for the creation of user-tailored stories on demand, in near-real time. (Note that although story and narrative are not exactly synonymous, we use the two terms interchangeably in this paper.) As a result, automated story generation has emerged as an important area of research. However, incorporating the human creative process involved in story creation into a computer program is difficult. The general goals for computer generated stories include novelty, coherence, and retention of reader interest. *Authorial burden* – how much effort is required from a human user/author for the program to be able to produce appropriate stories – is also an issue for creating stories automatically.

The literature also contains other terms that relate to storytelling. *Interactive storytelling* is an art form that seeks to include participation of users in how a story is told. While different people seem to have a slightly different definition of what is interactive storytelling, the main objective of this form of storytelling is to tell stories that are not *linear*. That is, they use inputs from the players acting as actors to change the direction and possibly the ending of a story – in effect, making them non-linear. Interactive storytelling does not require automated storytelling as we define it in section 4 below, but it can certainly include it. In this context, it is most similar to our interactive systems reviewed in section 4. On the other hand, interactive digital storytelling is an established research community all on its own with an annual conference, the International Conference on Interactive Digital Storytelling. The spectrum of research topics in this field tends to be much wider than the relatively narrow focus we have for the works covered in this review. Nevertheless, the intersection is found in the works we review under interactive narrative generation systems. Spierling (2005) contains a very good, although by now somewhat dated, description of the background and history of these two terms and their corresponding research communities. In discussing interactive storytelling, she states 'In fact, in games, as well as in constructivist scenarios for learning through a gaming simulation, users need to experience agency within a dramatic entity, and their roles change from being "members of the audience" to "participants".' (Spierling, 2005, p. 7).

The earliest reports of attempts to automate the generation of stories began to appear in the 1970s. Colby (1973) was the first to do so back in 1973. This work sought to create a partial grammar that could be used to create outlines of plots in Eskimo folk tales. In 1975, Rumelhart (1975) reported work to identify syntactic rules to summarise a simple story. However, neither of these works was complete nor could rightly be called an automatic story generator per se. That distinction belongs to TALE-SPIN (Meehan, 1977), which followed a few years later. It generated stories by using pre-defined rules that characters in the story could follow and act out. Ani (Kahn, 1979) followed TALE-SPIN shortly afterwards. Ani uses rules, knowledge, and constraints to create a 'film' (i.e., an animation) that describes a story being told, albeit in a very simple graphical format. The inputs to Ani are a high-level description of the characters to be in the story, the relationships among one another and the scenes in a Cinderella-type of story. The stories had to be interpreted by the human 'listener' based on observing the movements of the graphic figures in the story (e.g., a star, a square, a triangle), each of which represented a character in the story. Many other works have followed these seminal papers since their publication.

This paper is an extensive review of reported works that have had significant influence on the research discipline of automated story generation. The large majority of the works reviewed are used in entertainment, with only a few used for education. While we review some works used to create stories specifically for computer games, most of the works reviewed that are used for entertainment apply to a more traditional type of storytelling, where a story can be read or be told by a storyteller.

As we the authors reviewed the literature in the process of writing this paper, we were impressed and amazed by the richness, depth and creativity of the research reported. We review several important research works in automated story generation reported over the past 30+ years, but we specifically concentrate on works that appeared in the literature during the past decade (2010 to 2020). However, we should note that, unfortunately, this is not an exhaustive coverage of all works reported since TALE-SPIN and Ani, as the literature is quite rich and extensive, and an exhaustive coverage would quickly become intractable. Our sincere apologies to those authors whose worthwhile works are not reviewed here.

Finally, before continuing, we would like to say that the stories generated by most of the systems reviewed here are fictional, and most of them involve some level of fantasy. The only exceptions are Terminal Time (uses historical facts) and Crystal Island (used to teach biology). There is no attempt in any of these systems to write current news stories based on facts.

Structure of this Paper

An important consideration for any review paper is how to structure the paper such that the reviewed works can be grouped in meaningful ways, and that insights can be gained from examining trends in the reported works in one or more specific threads of research. Of course, there is more than one way to group the various works reported in the literature. We elected to group them using the criterion of whether the system being reviewed is by nature *interactive* or *non-interactive*. We next explain why we did so.

Non-interactive narrative generation, as its name implies, does not take any contributions from a human author while in the process of generating the stories, and the characters in the resulting stories cannot be controlled by human players as the stories unfold. The advantage, of course, is that it needs not coordinate the elements of the story with any user inputs. Conversely, the main disadvantage is that it must supply the creativity, understanding and world knowledge all on its own – a true Artificial Intelligence (AI)!

As the name suggests, in interactive narratives, the user is brought into the narrative generation process, allowing him/her to collaborate on the creation of the story. Interactive narratives can also have the user participate in the unfolding of the story itself, as if s/he were one of the characters in the story. In some cases, such participation is optional, while in others it is required for the story to proceed. The inclusion of the human element in the story generation or in the storytelling process fundamentally affects how the narrative generation system works, as it makes it a collaborative effort between human and machine. Such collaboration can facilitate the process of creating the narrative, as the human player/author can inject some of the human qualities that we still find difficult to reproduce in a computer, such as creativity, understanding, as well as account for the vast amount of world knowledge that we all possess. However, interactivity also introduces some difficulties, as now the automatically-generated story must be consistent and cohesive with what the human author/player contributes. Young (2000) discusses the compromises to be made between automated planning and human interaction. Riedl et al. (2003) expanded on this idea and built the Mimesis system to incorporate the approaches taken to accomplish this interaction successfully. A third type of system, we refer to these as *hybrid* systems, are those that can accept human involvement but do not require it. They have the best – and worst – of both worlds. Campfire is one such system and is reviewed below.

While interactive and non-interactive systems share the major goal of creating a story, they are fundamentally different in how they are enjoyed by the users. Non-interactive stories are enjoyed by simply listening to (or reading) the story passively. That is, the enjoyment is found in the plot, the characters, their evolution, and the story world. Interactive systems on the other hand, provide active enjoyment, where participation becomes the main vehicle for their enjoyment. This is particularly true for video games, but can also be true in less active narratives where the user plays the role of protagonist and makes decisions at various points in time as the story unfolds in real time. Given this fundamental difference, it becomes difficult to cross-compare systems belonging to a different one of these categories. Therefore, the research described and discussed in this paper is divided into non-interactive and interactive story generation systems. Computer (video) games are considered a subset of interactive narratives; however, we treat them here as a (smaller) third class for review purposes. We present non-interactive systems first because they can be in some ways simpler given the absence of unpredictable human inputs.

Section 2 below discusses various aspects of story generation and defines several technological concepts used in the reviewed papers. Its content will assist and inform the reader when s/he reads the subsequent sections.

Sections 3 and 4 cover non-interactive (section 3) and interactive (section 4) story generation systems. One sub-section of section 3 briefly covers systems that build narratives for video games. Each of these major sections has a sub-section that discusses the trends identified in the reviews and any insights gained from these discussions.

In section 5 we review the methods used by several of the authors to evaluate their systems and/or techniques. Such evaluation is important in any research endeavour and establishing a commonly-accepted and standard evaluation process and set of metrics can be extremely valuable. Finally, section 6 makes final conclusions.

Discussion on story elements and composition

In this section we discuss some of the background issues related to story generation. Here we define some terminology and briefly discuss some of the technologies used by some of the works reviewed. This will better prepare the reader to take full advantage of our more extensive reviews in sections 3 and 4. We begin the discussion by examining the four major aspects of a good story.

There are four major elements of stories, whether generated by a computer or authored by a human:

(1) The plot, or *fabula*[1] – a storyboard of sorts that stakes out the 'path' of the story.
(2) The characters – protagonist, antagonist(s) and other major supporting characters.
(3) The story world – where the characters live, work and act, and where the story is played out.
(4) The presentation of the narrative, or the *sjuzhet*[2].

In the next four sub-sections, we briefly discuss each of these elements and describe some of the techniques used to build stories.

Creating the Plot of the Story

Briefly, the *plot* is the sequence of actions and events that take place in the story. In other words, the plot is what happens in the story. The events that make up the plot affect how the state of the world changes as a result of intentional actions taken by the characters, or alternatively, by events beyond their control, such as natural disasters, accidents, weather, global conflicts, economic upheavals, epidemics, etc. The plot has the greatest influence on how interesting and enjoyable a story is; therefore, much attention has been focused on creating logical (i.e., causally linked), believable, and interesting plots.

The events in the plot progressively move the state of the story world from what is described at the beginning of the story, to the end state (aka, *goal state*) that (in most cases) resolves the conflicts that arise during the story, though not always completely or satisfactorily. Oftentimes, the objective of the story is to return to the initial state after it becomes changed for the worse by either natural events or the actions of an antagonist.

The structural similarity between a plot and a *plan* was not lost on early (also on current) narrative generation systems researchers. A plan is also a sequence of events, tasks or actions that will transform the state of something from a problematic or undesirable initial state to a more desirable goal state that meets the objectives of the planning agent. For that reason, it is no surprise that many narrative generation systems employ some sort of automated planning technique to build the plots. This is particularly true for non-interactive systems, but also the case for some interactive ones.

Therefore, it stands to reason that the most common technique used for automatically creating plots has been classical AI planning[3] techniques. Meehan's own TALE-SPIN employed automated planning through Hierarchical Task Networks[4]. There are many other examples of planning-based narrative generation systems, such as those using *partial order planning*[5] (e.g., IPOCL (Riedl & Young, 2010), VB-POCL (Riedl, 2010), CPOCL (Ware & Young, 2011) and many others). These and several others are discussed in more detail later in this paper.

However, there is a subtle, yet critical difference between plans and plots. In plans, the tasks have to be feasible and put into some chronological order that meets any pre-conditions required of the tasks. A plot, on the other hand, has the added burden of making the plot interesting to a reader. Thus, the events therein must be logical (causally-linked) and believable to the reader, but also include conflict and surprising or suspenseful events that make for an enjoyable story. The tasks/ actions in a plot depend on temporally prior events, so there must be a causal thread in the sequence. They must be related to the characters that perform the action, and those actions must be intentional for accomplishing their objective(s).

These additional requirements have led to the use of innovative architectures for plot creation. Some of these architectures have employed variations of the classical AI models of planning. These include the adoption of Long Short Term Memory (LSTM) neural networks[6], as found in Plan, Write, and Revise (Goldfarb-Tarrant et al., 2019). Brenner (2010) reports a way to do continual planning throughout the story generation and execution process.

Haslum (2012) argues that *model compilation*[7] can be used to enable stories to build their plots through traditional AI planning models when it would otherwise have been difficult to create a plot for these story models with the traditional AI planning models. He proposes ways to accomplish this compilation in his paper.

Porteous et al. (2010) tackle plot generation by applying planning through state constraints rather than by satisfying a set of preferences. 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 authorial burden. They view this issue as non-cooperative multi-agent planning.

Other approaches to plot creation do not involve formal planning methods at all. Wade et al. (2017) used stochastic methods to select the next event from a set of possible and contextually-relevant ones, depending on the probabilities assigned to each potential action. These probabilities are calculated based on weights linked to the attributes of characters. The next event is determined as the story generation progresses along.

Hollister (2016) used context-based methods to select the next event. His concept, called the *Cooperating Context Model* (CCM), was used in his Campfire Storytelling system (Hollister & Gonzalez, 2019) to ensure that the next event was contextually consistent with the state of the story world.

Selection of the next event by analogy has also been done through *Case-based Reasoning* (CBR)[8]. In narrative generation, CBR served to generate plots by comparing the needs of the state of the story world with other stored stories. Riu (Ontañón & Zhu, 2010), MINSTREL (Turner, 1991) and an un-named system by Coman and Muñoz-Avila (2012) are examples of systems that apply CBR to create plots.

One major problem with many planning-based systems, however, can be they can lead to a lack of character believability – an important element of any story. Planning systems generate a list of actions that comprises the story plot; however, they typically do not consider the underlying motivations for the generated actions. This can result in narratives that lack depth and seem more like a loosely-tied sequence of actions (i.e., like a plan!) rather than an actual story. There are other issues that arise with planners attempting to reason about other characters, such as with the *ramification* problem[9]. This occurs because actions that have context-dependent effects must be extracted from the current state.

Character development

The second important element of a successful story – some say the most important – is strong character development. How richly the characters are defined and how they might evolve throughout the story can also have a significant influence on the 'interestingness' of a story. Many stories depend on richly-defined characters that evolve throughout the story. Characters and plots are strongly connected – one cannot be effective without the other. The characters, if well-developed, can strongly influence what actions they take to cause the next event in the story. Conversely, prior events in the story have a significant influence on the characters and how they might change throughout the story.

Early systems such as UNIVERSE (Lebowitz, 1985) and BRUTUS (Bringsjord and Fertucci, 2000) focused more on the characters in their stories as a way to make them (and the resulting plots) interesting. Chang and Soo's Social Planner (Chang & Soo, 2008) uses characters' mental models to influence the plot, which can then change their mental models after the actions are taken. Porteous et al. (2013) argue that representing the relationships among the characters in the story is more important than plot development, and that a robust set of relations among the characters can unburden the plot generation process. The plot details are generated instead by the character relations and resulting interactions, in turn creating narratives that more closely resemble dramas and soap operas. AESOP and fAIble, take pains to modify the attributes of characters so that these evolve as the story unfolds. The characters' attributes, including emotions, change as events take place that cause them to be modified. These works are discussed in detail below.

One way to endow characters with rich and complex attributes is to build and communicate the history (prior to the start of the story) of the principal characters, especially as they relate to each other. Such *backstories* can help explain conflicts between characters in a natural way. These backstories provide a reason for some of the actions taken by the characters. They can also serve as the foundation for flashbacks and flash forwards. As an example, Tom Clancy novels, such as *The Hunt for Red October* and *Patriot Games* excel at meticulously building the characters, including their extensive backstories. fAIble makes it an important point to build these backstories for its characters. More on this can be found during the discussion of these systems below.

Building the story world

While creating interesting plots and characters are essential to generating stories, the story world also plays a major role. This is particularly important in tales of fantasy that take place in worlds where magic and fabulous beasts exist that do not always follow the same laws of nature that we do. Rich story worlds can lead to more immersive story experiences and increase the enjoyment of the narrative. J. R. R. Tolkien's *The Hobbit* and his subsequent *Fellowship of the Ring* trilogy are excellent examples of extensive world building in stories.

One way to enhance story worlds is through the characters that inhabit them. In video games, such characters are denoted as *non-player characters*, or NPCs. NPCs can play major or minor roles in plots, or merely exist in the background to provide realism, as would extras in a conventional movie set. While little creative effort is typically put into background characters that do not actively engage in story plots, having unique and believable NPCs can lead to better stories by creating more immersive worlds that a user can explore.

Presentation of the story – the discourse

How the story is communicated also makes a big difference. A well-written story, just like a well-acted play or a well-played piece of music, can significantly add to the enjoyment of a reader. Given that the most common way to represent a story is in a text narrative, efforts to improve the natural language used in the narrative were undertaken by several of the authors reviewed here. They consider this an important aspect of a story, but also a difficult one, as natural language generation is far from a mature technology at this time.

There are other ways to communicate the story. The Campfire and the fAIble systems explored alternative ways of delivering the story to listeners. Campfire used a *storyteller*, a lifelike avatar that strongly resembles actual people, and which reads the story text out loud through a *text-to-speech* (tts) system. Campfire also uses some rather primitive graphics alongside the story text to gain the interest of young children – its target audience. fAIble also experimented with lifelike avatars, but with less success than Campfire because of the lower quality of the avatar and the tts system used. fAIble did propose an interesting second means of presenting the story – a book that has the story text printed on its pages, which can be easily flipped by the listener to reveal the continuation of the story on the next page. This avenue opens several other ideas, such as a magical pen that writes the story text on the book pages in real time as they are read out loud by a tts system.

Graphical descriptions of the story are used by several of the systems reviewed. Picture Books and its successor Picture Books 2 allow a child to use graphic objects (dogs, cars, people, etc.) to trigger the system to build narratives around those objects. Conversely, Ani used simple graphic objects (star, triangle, circle, etc.) to play out the created narrative.

Nothing for Dinner (Szilas et al., 2015) presents the story through a video game interface. Through this interface the user can explore their story world as the protagonist and interact with other characters as they build the narrative. Crystal Island also uses a graphics-based, video game-like interface.

We now move on to detailed reviews of non-interactive narrative generation research.

Non-interactive narrative generation

Non-interactive systems focus on generating stories that do not permit (or require) human input during the narrative creation process. The focus instead is on the generation of interesting and novel stories that will hold a listener's attention and result in his/her enjoyment of it. A story ontology (aka, a *knowledge base*) typically needs to be incorporated manually into the system beforehand, as it is impossible to craft a story without background knowledge of the world – just like for human authors. Moreover, some user inputs may be required just prior to runtime to specify settings for the desired story, such as for example, the genre of the story, the projected length of the story, selection of characters, weapons used, and such. However, once the story generation process begins, user input is no longer necessary or even permitted. These types of systems are generally aimed towards entertainment and sometimes used to explore the automation of literary devices and styles.

One of the main issues tackled with these types of systems is how to incorporate the creative process necessary to generate interesting narratives. Interactive narrative generation systems can use human input to guide and move the story generation process along, but without such interaction, a non-interactive system must come up with the events of the stories all by itself. This can involve having access to a rich knowledge base that can be referenced for themes, ideas, and actions that make sense. The stories generated should be creative and interesting but not repetitive. Thus, the non-interactive systems need to be able to automate the creative process, but this is much easier said than done.

There have been several narrative generation systems that have paved the way for the recent advances in non-interactive narrative generation. The following section highlights some early (pre-2010) notable research systems that impacted this field.

Historical overview of non-interactive narrative systems

As mentioned earlier in this paper, TALE-SPIN (Meehan, 1977) was the first true narrative generation system reported in the literature, followed two years later by Ani (Kahn, 1979). Next came the UNIVERSE system (Lebowitz, 1985) that was designed to generate plot outlines for soap opera stories that could continue indefinitely (as soap operas tend to do). These stories were based on the interactions between characters and it pieced together pre-existing 'plot fragments' that fit the characters.

MINSTREL (Turner, 1991) used case-based contextual reasoning to generate stories about King Arthur and his knights. It used a Transformation Recall Adapt Method (TRAM) that allowed the system to alter events to fit a current story. The MINSTREL system is briefly discussed later in the context of MINSTREL Remixed, which replicated the original system and modified it to become interactive.

Bailey (1999) proposed an unnamed heuristics-based storytelling system that uses a reader model – what the reader was likely to be thinking – to guide the plot generation. It employed a heuristic search to select a next story segment from a predefined set that best met the reader model's needs.

BRUTUS (Bringsjord & Fertucci 2000) expanded storytelling by allowing the creation of stories across domains. BRUTUS used rules to generate stories that revolved around themes of betrayal and deception. The stories produced were grammatically correct and had a wide range of descriptive words, but required large amounts of pre-authored knowledge.

MEXICA (Pérez y Pérez & Sharples, 2001) introduced a very interesting concept that used an alternating engagement and reflection cycle to produce stories about ancient Mexican civilisations. The system first generated a series of actions and then reflected on them to incorporate additional details to the story.

STORYBOOK (Callaway & Lester, 2002) focused mainly on the natural language generation of narratives. It created stories for Little Red Riding Hood using a finite state narrative planner. However, all possible stories generated by STORYBOOK needed to be pre-authored, as STORYBOOK only sought to improve the natural language of those stories.

MAKEBELIEVE (Liu & Singh, 2002) uses fuzzy inferences to generate short creative stories based on causal chains. The system is interactive only in that it allows the user to input the first sentences of the story. From there, it breaks the sentence down into a cause and effect verb-object that is then linked to a causation sentence from the Open Mind Commonsense Knowledge Base (Singh, 2002). The sentences are linked via a fuzzy matching heuristic that uses lexical semantics to compare how closely the verb and objects are. The story is then built out in this way, by matching an event effect to the cause of the next event.

Chang and Soo (2008) implemented the Social Planner that uses rules that consider the mental states of other characters when generating actions. This approach is designed to address the issue of lack of character believability in plan-based systems. The plans are based on the main character's perspective, but the planner also reasons about the motivation and emotions of other characters. The actions performed by characters trigger emotions and the actions of other story agents. The social planner and its extra knowledge provide the system with the ability to generate plans that conventional story planners cannot replicate.

There have many other early systems that contributed to the non-interactive narrative field that have not been mentioned. This summary is not meant to be exhaustive but rather, to provide a brief historical backdrop of research in this area.

Recent non-interactive narrative creation systems

This section provides a review and discussion of more recent state-of-the-art non-interactive narrative systems and research from the past decade.

IPOCL

We begin this section with the IPOCL system, which is a modification of traditional partial-order planners (Riedl, 2010). These consist of a partially ordered set of events that make up a fabula. The partial ordering allows for events to be unordered relative to each other while maintaining a general chronological ordering so that the causal links of events are maintained. The Intent-based Partial Order Causal Link (IPOCL) planner by Riedl and Young (2010) incorporates motivation behind character actions to make the generated narratives more understandable and believable. Riedl and Young propose that along with logical causal plot progression, character believability is a universal attribute of stories. They define character believability as the perception of characters as intentional agents by the audience. That is to say, the readers should be able to infer and predict character's actions based on their innate motivations. Traditional planners often do not consider this as they create a solution based only on the start state and end goal state. The goal state is essentially the author goal, which can be different from individual character goals. Why are the characters doing the things they do? What is their intention? If it is simply to meet the end story objective, then it might lead to a less enjoyable narrative.

The IPOCL planner uses a search-based technique to find solutions with logical and believable action sequences, which create the appearance of intentionality. It deviates from traditional planners by differentiating between character goals and author goals and searches for plans where the character goals are potentially distinct from the goal of the author. Characters are given intentions either at the beginning of the story state or can develop intentions as the plan progresses. IPOCL searches for a solution in a space of plans and agent intentions and uses a special reasoning process to test the character intentionality from the audience perspective. After all, it is not enough for characters to behave with intention; the readers must be able to observe it. Therefore, the plan structure is expanded to include information on the individual agent intentions so it can be incorporated in the final solution. The resulting plan is then converted to natural language.

Below is a sample story from the IPOCL planner:

There is a magic lamp. There is a dragon. The dragon has the magic lamp. The genie is confined within the magic lamp.

King Jafar is not married. Jasmine is very beautiful. King Jafar sees Jasmine and instantly falls in love with her. King Jafar wants to marry Jasmine. There is a brave knight named Aladdin. Aladdin is loyal to the death to King Jafar. King Jafar orders Aladdin to get the magic lamp for him. Aladdin wants King Jafar to have the magic lamp. Aladdin travels from the castle to the mountains. Aladdin slays the dragon. The dragon is dead. Aladdin takes the magic lamp from the dead body of the dragon. Aladdin travels from the mountains to the castle. Aladdin hands the magic lamp to King Jafar. The genie is in the magic lamp. King Jafar rubs the magic lamp and summons the genie out of it. The genie is not confined within the magic lamp. King Jafar controls the genie with the magic lamp. King Jafar uses the magic lamp to command the genie to make Jasmine love him. The genie wants Jasmine to be in love with King Jafar. The genie casts a spell on Jasmine making her fall in love with King Jafar. Jasmine is madly in love with King Jafar. Jasmine wants to marry King Jafar. The genie has a frightening appearance. The genie appears threatening to Aladdin. Aladdin wants the genie to die. Aladdin slays the genie. King Jafar and Jasmine wed in an extravagant ceremony.

The genie is dead. King Jafar and Jasmine are married. The end. (Riedl & Young, 2010)

The sample story clearly demonstrates character motivation. While the sentences are simple, the plot is not just a sequence of events as there is clear intent behind each character's action.

One of the main limitations of IPOCL was that it was built so that each character goal was achieved, which limits several possible narrative paths. Character failure can result in just as compelling narratives as those in which they succeed. Nonetheless, the IPOCL planner represents a significant milestone in narrative generation.

VB-POCL

The Vignette-Based Partial Order Causal Link (VB-POCL) (Riedl, 2010) searches for a story among alternative stories (the vignettes) to find the best story plan. It does this through a refinement search, in which a plan is chosen, inspected for flaws, and is revised to remove any flaws found by replacing them with events taken from the matching vignettes. However, stories consist of more than just a coherent series of events, as world and character building are important factors that can make the story more relatable and enjoyable. As such, stories may often contain events that, while not necessary for the plot, add to the creativity and depth of the narrative.

VB-POCL uses vignettes – story fragments – that represent good examples of specific situations that can occur in stories. In a way, they are an encoding of the creative knowledge for events about which the system would require specialised knowledge in order to reproduce. The vignettes are stored as partially-ordered plan fragments. The planner then uses the vignettes as actions that provide the story with more detail and lead to more aesthetic narratives. The system also uses vignette transformation to generalise vignettes from one story domain to another, leading to cross-domain use of vignettes. This process resembles Case-based Reasoning in several ways.

The major drawback of the VB-POCL planner is the lack of resource libraries for vignettes and story domains in a computerised form. This limits the system because a large pool of vignettes would be necessary to create diverse stories. It also increases the burden on authors who would need to provide the knowledge base to the system. Overall though, this system takes an interesting approach to increasing the creativity of stories without the need to code the logic necessary for specialised events.

CPOCL

Interesting stories usually have conflicts that create the basis for the plot. For example, a boy wanting to propose marriage to a girl, and then doing so would be an acceptable plot, albeit simple and rather boring. However, a story of two boys wanting to propose to the same girl creates conflict, which in turn results in a more engaging plot. The CPOCL planner (Ware & Young, 2011) addresses how to generate conflict in stories without compromising the soundness of the plot.

A story plan is defined as the sequence of steps that must be taken to get the story from the start state to its end state. In the conflicting proposal example story above, the start state would be for one of the boys to have a ring while the goal state would be to have the other boy (without a ring) successfully propose and marry the girl.

While an easy solution to including conflict in automated stories would be to have the story generated from pre-authored conflict scripts, this can result in problems with the soundness of the narrative. This is because the search for a plan that resolves the story occurs without explicitly reasoning about the provided conflict. In a simple planner, the shortest plan solution would have the boy with the ring give the ring to the other boy so that he could successfully propose. This, however, does not make logical sense and would lead to a boring story.

CPOCL addresses this by considering plans for each individual character and how these plans can interfere with one another's goals. Instead of allowing all pre-conditions for a character's plan to be met, the system will preserve certain threats to the plan and leverage this to create the story. These result in conflict, which can be internal (when a character thwarts its own plans), external (when another character thwarts another character's plans) or environmental (as a result of a state incompatible with the desired goal). For example, an external conflict would result in the boy with the ring losing it and the other boy finding it and proposing to the girl.

Limitations on this system arise when considering whether to introduce conflict in generated stories that do not naturally require conflict. This can cause the story to feel forced. The system also does not allow for characters with failed plans to create a new path to the same goal or switch goals. Nevertheless, CPOCL demonstrates how plans that would otherwise be ignored by traditional planners because of unresolved character plans can result in interesting and diverse plotlines for generated narratives.

Un-named system by Niehaus and Young

An un-named system by Niehaus and Young (2010) created narratives that required the reader to infer the details of a story. Inferences are pieces of information that readers use to improve comprehension of a narrative. They can make stories more interesting by making readers think about the narrative rather than just reading a sequence of events. Inferences are a type of participation on the reader's part by letting them fill in missing details in narratives. In stories that lack certain key details, a user must infer those details; however, leaving out essential information can also lead to story incoherence. Therefore, there must a balance between the information provided to the user and the information that is left out.

Niehaus and Young modified the IPOCL story planner to generate stories that prompt such inferences from the reader. In their algorithm Niehaus and Young use a computational model of Constructivist Theory[10] and partial-order planning to generate their narratives. The approach implemented in this system required the human author to provide the goals and a rough plan in order to initiate the story generation process. Once a plan has been generated, 'holes' in the narrative are created that lead the reader to make an inference about what happened. The planner provides just enough information to narrow down the possible inferences and allow the reader to jump to the intended inference. Criteria for causal and intentional inferences are used to search through a list of partial plans. Then the inferred steps are removed for the final output story.

The authors distinguish between two types of inferences: causal inferences and intentional inferences. Causal inferences are those needed to maintain causality, such as when a world state changes without explanation. Intentional inferences are those needed to fulfill a character's intention, such as when a character has a goal and the audience infers that it will act to accomplish its goal.

An example narrative from the system is shown below:

"Brian had been famished since he started his new, healthy diet.
The waiter handed him a menu.

At the top of the menu was a bacon cheeseburger." (Niehaus & Young, 2010).

In the short story snippet above, the reader can infer that Brian is in a restaurant. Likewise, the reader can infer that the bacon cheeseburger represents a powerful temptation to a hungry person.

The main problem with the types of narratives generated by this algorithm is that some readers may not successfully conclude what was intended, leading to story incoherence in their minds. The reader must understand the context of the story, the character's intentions, and agree with the author's rules of the story world. Therefore, there must be an understanding of these things before reading the story. However, this modification of the IPOCL system does produce interesting narratives that fulfill the goal of involving the user in a non-traditional way.

Un-named system by coman and muñoz-avila

Coman and Muñoz-Avila (2012) created a system that, like Riu and MINSTREL, used a case-based planning[11] technique that could generate diverse stories based on a single common starting scenario. Unlike the latter two systems, this one is non-interactive. The stories generated could share the same ending or have alternate endings, but each story would be composed of a different sequence of actions. This was done to address the concern about lack of diversity in story plots from generated narratives, as diverse stories are the key to maintaining user attention and entertainment over the long run.

The plans in their system are composed of an ordered array of actions that link events together to make up a storyline. The system has a case base that contains a collection of varied story plans that it can reference. Given a starting scenario and a desired ending, the system first uses a case retrieval technique to find a case that best matches the input. The case retrieved must start with the given scenario and result in the end goal, unless no end goal is given. Next, an adaptation procedure is used to modify the retrieved case so that it fits with the story scenario. A *plan distance* metric is used to determine whether storylines are meaningfully different. A modification of a formula widely used in recommender systems then uses the distance metric to define the relative diversity between plans. This process is repeated to produce diverse story plans.

For example, if a story involves a peasant who has to rescue a friend, then the starting scenario would be having a peasant protagonist who has lost a friend, and the ending would be rescuing the friend. The system would then pick a case from the case base where the acting agent is a peasant. Suppose that the retrieved case has a plan where the peasant collects a nearby treasure and then attacks an ogre to save its friend. The system can then pick a diverse storyline where the peasant cuts down some trees to rescue the friend. The first story can be characterised as reckless and greedy while the second one is characterised as intelligent but environmentally damaging. The two stories are different and the character uses different approaches to meet its goal, yet the overarching plot is still the same. If no ending had been specified, the system might have chosen a plan where the peasant collected the treasure and then went off to find another adventure instead of saving its friend.

The system's strength lies in its ability to re-use plans for new stories. This can reduce the amount of computation that it takes to produce story plans. However, the key for this system to work is to have a suitable case base that contains a diverse collection of plans. Without this, the algorithm behind the story generation will not work properly. Coman and Munoz-Avila recommend using first-principles planning[12] techniques to create plans that could populate a case base.

Dramatis

O'Neill and Riedl (2014) developed Dramatis as a computational model of suspense for storytelling. Suspense is a popular element in stories. It creates feelings of apprehension for the story protagonist as it faces possible failure and keeps the readers engaged in the story. As a result, the inclusion of suspense in computational narratives would allow for more creative and entertaining stories. The Dramatis system reads in a representation of a story and evaluates the suspense it elicits. It doesn't actually generate stories, so it is not a narrative generator as such.

Dramatis measures suspense using a psychological perspective that states that the feeling of suspense is created when a protagonist's avenues of success decrease. The suspense is correlated with the perceived chance that the protagonist will succeed. A memory model is used by the system to track story information in a reader's conception. The Modified Event Indexing with Prediction (MEI-P) component is an activation network that holds story elements and denotes important elements as being more active. Elements in the reader model with high activation levels represent information that a reader recalls fastest. This model is used in the generation of 'escape plans', outcomes in which the protagonist avoids failure, along with a modified planner. The escape plan's quality is used as an equivalent measure of the suspense felt by the reader throughout the story. Testing showed that both Dramatis and human readers produced the same ordering of stories based on the suspense level elicited.

In summary, Dramatis begins the work of exploring suspense in computer-generated narratives. It could serve as a tool for producing stories that are as interesting and engaging as those written by human authors. While the current version only evaluates suspense, it could conceivably be implemented in a narrative generation system to interactively change the story plan to be suspenseful. This would add to the richness and enjoyability of computer-generated narratives.

Flux capacitor

While not a full-fledged narrative generation system either, the Flux Capacitor is a transformative arc generator that creates story pitches for interesting and dramatic stories (Veale, 2014). A *story arc* is a plot line in a narrative that usually follows a specific character and the transformations it undergoes as the story progresses. Character transformation stories are classic throughout storytelling history. They can be seen in the hero's journey and in popular media (e.g., television series such as *Breaking*

Bad and *Imposters)*. Watching the protagonist change drastically can be exciting, as the audience watches the nuanced transformation. The Flux Capacitor automates the generation of such arcs so that they can be used by narrative generation systems to generate compelling narratives.

Transformative arcs are composed of a start state, an end state, and a blend of states that incrementally lead from the start state to the end state. The Flux Capacitor works by using corpus analysis to acquire knowledge for generating and picking these states. The corpus used by the Flux Capacitor is Google n-grams that use over five million books as its source. Google n-grams is also used as a database, and a model of world-knowledge, to create an arc composed of character states linked together through conceptual integration networks. The main goal is to pick opposing starting and ending states because this leads to conflict that can add richness to narratives. The transformation between the states will lead to conflicts between the traits and choices that a character makes as they change. The result is a dynamic blend of properties that binds the start and end states. The final output is a concept story idea that uses natural language to deliver a story pitch.

The snippet below is an excerpt of a story pitch about how a nun slowly becomes a prostitute.

Nun condemns chastity, wallows in wickedness.
Nun fatigued by fidelity, veers towards vices.
Nun goes from being unflinchingly faithful to being increasingly unfaithful.

Nun goes from wearing wimples to wearing hotpants. (Veale, 2014).

This story pitch can then be used by a storytelling system to generate a full narrative depicting the transformation of the character and the reasons for the transformation. The Flux Capacitor provides information about the transitions and relationships between the states for the storytelling system to utilise during the narrative generation.

The Flux Capacitor sets the groundwork for creating transformative character stories. While the transformations it generates are certainly dramatic and intriguing, it only produces a story pitch that lacks the reason behind the character's transformation. In collaboration with a narrative generation system, the states produced by the Flux Capacitor could be used as landmarks that guide the narrative development. The narrative system would need to produce a reason behind the character's transformation though, and a conclusion about their change.

MEXICA-Impro

MEXICA-Impro (Perez y Pérez, 2015) is an expansion on the original system MEXICA (Pérez y Pérez & Sharples, 2001) for creating a social virtual world in which virtual collaborators can interact to create better narratives. This is accomplished through two agents working together to produce stories that neither of the agents could replicate if it worked alone. The narratives produced by such a system are thus referred to as *collectively-creative stories*.

The two agents used in MEXICA-Impro are instantiations of MEXICA, a narrative generation system that creates stories about the Mexica people of Mexico. MEXICA works by using a knowledge base populated by the user and that contains a dictionary of actions, each with preconditions and consequences, as well as a set of previous stories. Through this information, MEXICA produces stories through a contextually-based system, and its knowledge base changes and expands with the new stories it creates. It does this through an Engagement-Reflection cycle. During the engagement cycle, a plot is created through a sequence of actions but the preconditions are not checked. In the reflection cycle, preconditions are verified, and actions are added to fulfill them, novelty and interestingness is evaluated, and the system attempts to resolve any impasses.

MEXICA-Impro has the two agents configured differently so that even though they are based on the same model, they can each contribute differently to the story. One instantiation is called the Leader, which begins the generation process and goes through an Engagement-Reflection cycle before signalling to the second agent, the Follower, to continue developing the plot. The agents then switch control back and forth after every cycle until the story can no longer be extended. Agents are not allowed to modify each other's output or their own after their turn. Once the story is completed, the actions are replaced with predefined text.

Below is an example of a story generated by MEXICA-Impro:

> *"Jaguar Knight was an inhabitant of the great Tenochtitlan. Virgin was an inhabitant of the great Tenochtitlan. From the first day they met, Virgin felt a special affection for Jaguar Knight. Although at the beginning Virgin did not want to admit it, Virgin fell in love with Jaguar Knight. Jaguar Knight felt a strong attraction for Lady. A bad spirit took Virgin's soul provoking Virgin to become intensely jealous of Lady. Virgin could not bear Lady's arrogance! Virgin's frame of mind was very volatile and without thinking about it Virgin charged against Lady. Virgin hated Jaguar Knight. Virgin was emotionally tied to Jaguar Knight but Virgin could not accept Jaguar Knight's behaviour. What did Virgin do? Virgin felt deep odium for Jaguar Knight. Invoking Huitzilopochtli, god of the war, Virgin cut Jaguar Knight's jugular. The blood covered the floor. Lady thoroughly observed Virgin. Then, Lady took a dagger, jumped towards Virgin and attacked Virgin. In a fast movement, Virgin wounded Lady. An intense hemorrhage occurred which weakened Lady. Virgin felt panic and ran away to hide in the Popocateptl mountain. Virgin was walking when Ehecatl (god of the wind) blew and an old tree collapsed badly injuring Virgin."* (Perez y Pérez, 2015)

The sample story above sounds realistic and it is interesting to read. It provides a semblance of a description of the action, which adds richness to the narrative. The story has a well-defined beginning, but at the end, does not conclusively indicate Virgin and Lady's ultimate fate – do they die? Or do they survive and return to Tenochtitlan to continue their conflict?

Story actions are replaced with predefined text, which means that for every action a human developer must write the corresponding sentence. This represents significant authorial burden and increases the time it takes to expand the system to include new story actions. Overall however, MEXICA-Impro takes an innovative direction in narrative generation research. Using more than one agent in story generation could lead to more creative and interesting stories than those generated by a single agent. The downside to this system is that the natural language generation is not dynamic.

(1) **HEADSPACE**:

HEADSPACE (Thorne & Young, 2017) creates stories that track and manipulate character's beliefs about their world. Many narrative generation systems select actions that will lead the story to a final goal. Characters will perform these actions and move the story forward. Usually, though, the character's reasoning behind their actions isn't considered. Moreover, failed actions are generally avoided to prevent the story from heading to a dead end or to an unresolved goal. However, characters failing in stories are a common occurrence and often enhance the narrative. Human authors often include failure in their characters in order to build suspense in stories, increase character development and prolong the attainment of the final goal. Failed actions in stories occur because the character incorrectly believed that her/his action would be successful and help it get close to its goal. Traditional plan-based stories produce many actions but lack the range of features that are prevalent in human produced narratives.

HEADSPACE implements a belief model that works with a state-space planning system[13] to build story plans that allow characters to fail. These failed actions are explicitly planned based on the knowledge limitations that characters have about the story world. Specifically, characters in HEADSPACE stories can operate under mistaken beliefs that cause them to act and fail; the characters observe the result of their failed actions, and consequently respond by revising their beliefs.

Actions in the system are represented through their preconditions and effects. The effects are the story world conditions that change because of an action being executed. There are two main action categories: environmental actions and character actions. Environment actions are those performed by the story world, such as a character being struck by lightning. The system also categorises the *story world literals* – story states with true or false values. *Material literals* refer to physical world states; *epistemic literals* are those that a character believes, whether correctly or not.

Actions fail because of preconditions not being met. These are referred to as *inexecutable* actions. When a character attempts to perform an inexecutable action, the planner follows a simple failure policy. First, the effects of the intended action are not implemented, and no material world conditions change. Second, the character who attempted the action immediately notices the failure but no other character does. Lastly, the character assumes that the failure was caused by the epistemic actions of the precondition not holding in the current world state.

For example, a character locked in a cell wants to escape. If he has a gun that he believes is loaded, he will attempt to shoot the lock on the door. However, if the gun was unloaded then this action clearly fails. The character will realise this and will question his beliefs about the preconditions involved in the action. The character will check the gun and consequently update his beliefs about the current story world state.

HEADSPACE's planning system introduces a novel method for including simple suspense building in stories. Allowing characters to fail diversifies the actions that can occur and facilitates the generation of more compelling and natural sounding narratives. Some current limitations of the system include the scope of preconditions questioned after a failed narrative and the lack of failure awareness of other story characters. When a character questions its beliefs, this extends to all preconditions of the action. In the example mentioned, the character would not only question the gun being loaded, it would also question the location of the door. In this case, the location of the door is irrelevant to the failed action but the system has no way of knowing this beforehand. Characters that are in the same location as the character that attempts an action should also be made aware of the failure, as they share the same story space. Nonetheless, HEADSPACE effectively implements a model for action failure based on incorrect character beliefs, and the subsequent revision of those beliefs.

Provant

Porteous and Lindsay (2019) describe this system as a counter planning approach to plot generation. Their main objective seems to be to reduce authorial burden, but it is an innovative approach all in itself. The main concept behind this is that the plot centres about the story's protagonist and its objective, but it faces multiple serial obstacles throughout the story as it seeks to accomplish its objective(s). These obstacles are put in place by the antagonist, whose objective is to keep the protagonist from accomplishing its objectives. The protagonist, of course, needs to overcome these multiple obstacles, one at a time, in order to reach its goal state. The authors refer to this process as non-cooperative multi-agent planning and *counter planning*. The authors further indicate that there is one major difference between the mainstream counter planning research and its application to narrative generation: the obstacles put up by the antagonist must be *recoverable* by the protagonist – that is, there must be ways to overcome it or get around it. Otherwise, there may not be any story to tell.

One interesting aspect of their work is that the protagonist's goal is not necessarily known explicitly by the antagonist. Instead, the latter has to infer it by observing the behaviour of the former. The protagonist's goal is typically composed of sub-goals, the achievement of which will get the protagonist closer to its overall goal. The antagonist is designed to block these intermediate objectives along the way as the story unfolds.

Provant is sub-divided into several narrative segments. The first of these is called the Exposition, where the protagonist and antagonist are introduced, and their objectives are described. It sets the stage for the story. This is then followed by an arbitrary number of so-called *episodes* that narrate all the actions produced in the protagonist vs. antagonist conflict. If the protagonist succeeds in overcoming all the obstacles presented by the antagonist, then the final segment (*Resolution*) plays out, where the protagonist achieves its ultimate goal.

The authors describe the problem in Planning Domain Definition Language (PDDL) (McDermott et al., 1998) and provide the algorithms in their paper. The system appears to produce the plot outline and not a complete, ready-for-telling story. The paper includes an extensive evaluation of their system.

(1) **Suspenser**:

Suspenser (Cheong & Young, 2015) aims to generate varying levels of suspense narrative by simulating the human reasoning process. Their approach to generating narratives is very unique. To better understand Suspenser, it should be noted that the type of suspense it aims to model is ' … *the class of suspense associated with the perceived likelihood of undesirable outcomes over preferred outcomes*' (Cheong & Young, 2015, p. 39). Put simply, the system wants to model the suspense that comes when the protagonist is left with a dwindling number of options and the reader is aware of additional information that the protagonist does not know. Input to Suspenser is given as a set of all story actions in the form of a large corpus of text. One of the key components of Suspenser is how it models the human reasoning process (referred to as the Reader Model). The authors rely on research that states partial order planning algorithms can effectively model human reasoning. Their planning algorithm Crossbow then aims to simulate which story route will result in the desired suspense level. The output of the reader model is sent to the suspense creator, which then rearranges the initial text document to create the final story.

The Suspenser system takes four inputs: a fabula, a suspense level (high or low), a goal, and a specific point *t* in the story plan, which represents the ultimate moment of suspense. Given these inputs, the system utilises three components to build out the narrative. The first is the Skeleton Builder, which differentiates between necessary events (kernels) and auxiliary events (satellites) in the fabula. The kernels are defined as events in the story that cannot be removed without sacrificing the soundness of the story. This is determined by computing the number of causal relationships an event has with other events. The top K most important events are then selected as the story kernels. All other events are regarded as satellites, or events that are not necessary but enrich the narrative. The resulting sequence of kernels is the skeleton.

This skeleton is then passed to the Reader Model which simulates the reader's reasoning process and checks the skeleton for coherence. It does so by considering the reader's knowledge at any point in the plan. If the skeleton is found to be incoherent, it is sent back to the Skeleton Builder, which will take the next most important satellite event and add it to the skeleton. This is done until a coherent sequence of events is found.

The sequence of events is then passed onto the Suspense Creator that predicts which elements of the story can be manipulated to elicit suspense from the reader. This is done by finding a series of additional plan steps that precede step t. The Suspense Creator uses a heuristic function that measures an event's suspense level increase by computing it as the inverse of the number of planned solutions to reach the goal that are available to the protagonist based on the reader's knowledge. This is based on the idea that as the number of possible preferred outcomes decreases, and the undesired outcomes increases, reader suspense is increased. These events are added to the plan to create the final solution.

IRIS

The IRIS system (Fendt & Young, 2014) is designed to plan plots that permit characters to change their plans as a result of what transpires in the story. This is called *intention revision*, and it is a natural occurrence in human-authored stories (and in the real world!). Intension revision can result from characters having conflicting goals, and one character finds that it can no longer achieve its goal. This is defined in a framework that describes the goals and intentions of the characters in terms of Bratman's Beliefs, Desires and Intentions theory (BDI)[14]. Each character in the story generated by IRIS will have its own BDI mental model that includes the plan to achieve the goals it selected at the beginning. The IRIS algorithm that produces these plans is a partial order planner called Longbow that was developed by Young earlier.

Intention revision occurs in the algorithm when a character's plan can no longer succeed in light of the state of the story. IRIS differentiates between plan revision and intention revision. The former is the change of plans but not of the character's intentions; in the latter, both the intentions and the plans change. Intention revision involves several steps and leads to a new plan to embody the new intentions of the character. The new intentions can be similar to the old ones (but not identical) or it can be motivated by revenge against the other character(s) that might have blocked the former plans.

Campfire Storytelling System

The Campfire Storytelling System (Hollister & Gonzalez, 2019) creates bedtime stories for young children with the plots revolving around a protagonist on a quest for an item who faces several perils on its journey. One of Campfire's innovative features is that it uses a context-centric approach where, rather than directly looking for the next event to take place, it looks for the next context to be identified (the current context could also remain unchanged). Then, within that context, it will select from the contextually-appropriate next events. This leads to logical and believable plots that are assured to be context appropriate.

Campfire uses templates of quest stories to guide the plot generation process. Given that all of its stories are of the quest genre, the plot at its highest level is composed of either five or seven stages – the five-stage template is used for shorter stories. Campfire is composed of four main modules: the *storyteller*, the *knowledge base*, the *director*, and the *user interface*. The storyteller is the lifelike avatar that uses text-to-speech to orally narrate the story to the child. The knowledge base contains all the information necessary to build and fill out the story world with actions, characters, and monsters. The director module is the most important and complex one. It works with the knowledge base to fill in the story outline using a *Cooperating Context Model* (CCM) algorithm (Hollister & Gonzalez, 2018). In Campfire, *contexts* are defined as groupings of knowledge about objects, characters, actions and expectations that are appropriate in particular situations. For example, if one is in a library context, context-appropriate objects would include books, magazines, tables, chairs, copy machines and computers. Likewise, contextually-appropriate characters would include reference librarians as well as other users of library services. Actions would include reading, performing searches, speaking to a librarian; expectations would be to be quiet and speak only in hushed tones. Contexts can also have objectives. The CCM process used in Campfire identifies the situation being faced by the character that is to perform the next action, and determines which contexts would be appropriate for that situation. It then selects the one context that is most appropriate and within it, the most appropriate of potentially several next actions. The CCM algorithm not only 'executes' a list of contexts that contain actions to perform in each situation, but it also generates the corresponding text that describes the actions in natural language. CCM is a variant of *Context-Based Reasoning* (CxBR).[15]

An excerpt of a story produced by Campfire is shown below.

> *" . . . On the way into the village, Billy spotted an elderly figure in distress on the trail ahead. Not wanting to get involved, the jedi went and hid until nothing could be heard. After thinking coast is clear, our cowardly jedi continued towards village. When he gets to the point where the figure was, the cowardly jedi discovered a dying farm hand. The farm hand softly spoke, "Please help me! I Cannot finish my lifework. An evil lord has identified the current location of the ring of power and you must find it first to protect it but be careful as the ring of power has the power to rule the world. Will you help an old farm hand and save the world from evil?" Billy first response was the same one he has given all of his life but something was felt different this time. Our cowardly jedi want to take personal responsibility for once in his lifetime and said, "Yes!" A Brief smile appeared on the farm hand face. He then spoke up again and said, "Head north until you reach a spooky, haunted forest. Continue into the forest till you find the foreboding, enormous temple. Within the temple you will find the ring of power. This book will help you in your journey. Now go!" . . .* (Hollister, 2016).

The sample story above is extended (i.e., actually longer than what is depicted above) and entertaining. There are some grammatical mistakes in the dialog, but it is nevertheless easy to understand.

However, when one compares multiple stories produced by Campfire, one can notice a repetitive underlying pattern in the quest stories produced. This comes from the use of the quest templates and results in lack of story diversity. Campfire also relies on large amounts of pre-authored knowledge to create its stories, leading to a significant authorial burden. Nevertheless, the stories Campfire produces demonstrate the potential of automated narrative generation for children's stories.

AESOP

Before we begin this review, we should note that unfortunately, there are two systems that share the same name. They are quite different from each other and created by different authors. More significantly, they are independent of each other, i.e., they do not build on one another. To minimise confusion, we refer to this one by its name in all CAPS (AESOP), while the second one, reviewed in section 4.2.6 below, is expressed in mostly lower case characters (Aesop).

The Automatic Eclectic Story Origination Program (AESOP) eschews the use of planning systems altogether for building the plot. Instead, it generates events in the story through a stochastic event selection approach driven by probabilistic logic (Wade et al., 2017). The event generation sub-system selects the next event probabilistically, and the likelihood of one candidate event to be selected is weighted by the attributes of the character that is to take the action. For example, when a character is threatened by a dragon, the candidate events could be to fight the dragon, flee or hide, among possibly other options. So, if the character so threatened happens to be highly skilled with weapons and it has a suitable weapon at hand, the probability of fighting rises over the others. Alternatively, if it is a weak fighter but fleet of foot, then the probability of its flight will be weighted higher than the others. In any case, the event with the highest probability is not always selected, thereby injecting some surprises to the events selected. AESOP can also use contextual reasoning to ensure that the action selected for the next event is contextually appropriate. It employs *contextual graphs* (Brézillon, 2004) to ensure such contextual validity. This combination allows for a narrative generation process that can generate interesting event sequences without needing large amounts of pre-authored material.

AESOP is composed of four fundamental structures: Characters, Conditions, Actions and Objects. These structures, or modules, make up the knowledge base used by the event generation process. Characters represent agents that cause the story to advance through their actions. These characters are defined by their attributes, relationships and goals. Attributes are properties such as strength and charisma that can affect the actions that the characters perform. Actions performed by characters are also influenced by the relationships they have with other characters in the story world. The relationships are represented by a positive or negative value as well as a direction denoting the receiver of the relationship. For example, a Character A can 'like' a Character B. Characters are also guided by their goal, which are represented as specific Conditions that a character would like to have occur. Conditions are states that represent the story space at any a given time and are used to keep track of the story world as it is transformed through actions. Actions make changes to the story world. Some of the properties of Actions include pre- and post-conditions as well as a probability. The pre- and post-conditions are used to define which actions can happen during a certain story world state, and how the story world will change after the action is performed. The probability is used to determine the likelihood of an action being chosen. This probability is recalculated throughout the story and is affected by the character attributes, which can also change throughout the story as a result of events. For example, a character who is strong is more likely to pick up a heavy item than would a weak character. Lastly, Objects are items with which characters can interact. These modules can be expanded to include new assets that increase the possible stories generated by the system.

The story itself is generated using the Expanded Story Space Model. The model has a *story space*, a *possibility space*, and the *current space*. The story space contains all possible actions in the story. It is built at the start of the narrative generation process by going through all the actions and permuting them with the relevant characters and objects of the story. This is then saved so that it does not have to be recreated again during the iterative event generation process. The possibility space contains all possible actions given the current story conditions (i.e., context). This possibility space is a subset of the story space and is derived through a running process that uses pre-conditions as the criteria – in other words, it contains the actions that are possible in the current context. Story actions are then chosen from the possibility space based on their probabilities and through a stochastic selection procedure. The chosen action's post-conditions are then used to update the story world accordingly. The action is then added to the current space, which is the fabula, and holds the sequence of actions performed so far. This process is repeated until a threshold number of actions have been reached or the character's goal is accomplished.

An example output story from AESOP is shown below:

> *"Witch reads Book. Witch meets Harriette. Harriette antagonizes Witch. Harriette becomes an enemy of Witch. Witch seduces Harriette. Harriette becomes a friend of Witch. Harriette asks for Witch's hand in marriage. Harriette eats Cookies. Witch agrees to marry Harriette. Harriette seduces Witch. Harriette befriends Witch. Harriette attacks Witch. Harriette kills Witch. Gary moves to Forest from RedHouse. Gary moves to WitchHouse from Forest. 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."* (Wade et al., 2017).

The above story sample, while far from perfect, accomplishes several goals. AESOP can produce (arguably) interesting stories that do not rely on computationally-expensive planning algorithms or a substantial amount of pre-authored material. Instead the story is driven by the characters' traits and the way they relate to the story world. However, there are several weaknesses in the system: first, the events in the story are short and lack detail – they could be considered a plot outline rather than a plot; second, there is little explanation for why these events happen or what their outcome is (e.g., Harriette attacks the Witch, but who wins is not stated); third, there is randomness in the sequence of events. For example, Harriette asks Witch to marry her and the Witch agrees, but these events are interrupted by Harriette eating cookies. This far less important event disrupts the story flow and does not add substance to the story. Fourth, the story weaves back and forth between love and hate among the characters, indicating lack of memory. Overall, even with these issues, the AESOP system shows potential for generating interesting stories using stochastic methods.

(1) **fAIble**:

The fAIble system (Alvarez et al., 2019; Bottoni et al., 2020; Kazakova et al., 2018) followed the automated story generation systems CAMPFIRE (Hollister & Gonzalez, 2019) and AESOP (Wade et al., 2017). Their primary objective is to generate children fairy tales as the output stories. fAIble uses a graph-based framework for storytelling that focuses on generating stories through sensible character actions. Traditional plot planning is avoided, as in CAMPFIRE and AESOP, because of its limiting effects on story diversity and scalability. fAIble implements contextual reasoning to avoid irrelevant story actions. It also minimises the use of templates to increase variety of stories – limitations that were evident in CAMPFIRE.

The fAIble system has undergone three iterations. The fAIble I system (Kazakova et al., 2018) is divided into three main components: a graphical database, an event generation component, and a natural language generation (NLG) component. The graphical database is used to store the story world information and *common-sense templates*. These templates are used to increase the narrative quality without the need for substantial amounts of pre-authored knowledge. The event generation component is then guided by these templates through a probabilistic selection process derived from AESOP to allow the possibility of unexpected story elements, such as a princess robbing a thief. While templates are used to guide the event generation, it is ultimately up to the character actions to move the story forward. These actions are generated using an 'Action-Thinking' graph that is an abstract representation of character reasoning. This provides a way to identify the cause of events. Dead ends are avoided using deus ex machina events to resolve the story if needed. Once an action or event is created, it is processed and realised into an English sentence using the SimpleNLG engine (Venour & Reiter, 2008).

The fAIble II system (Alvarez et al., 2019) builds upon the original iteration of fAIble and improves the event variety as well as makes the final story sound more natural. The fAIble II system focused primarily on character development as the driving force behind the story plots. The graphical database shifted from being used for decision making to only store the story state to allow for cascading relationships between the main character and other story characters and components. The story is then driven forward by the main character's actions which are influenced by that character's perception and emotions about the story world. Lastly, a visual representation of the story is created through the animation of the book discussed previously on whose pages the story text is printed. The main limitation found in this system was that while the characters acted based on their emotions and perceptions, the reasons for these feelings and ultimately their resulting actions were not adequately explained in the story text.

The fAIble III system (Bottoni et al., 2020) was the final iteration and built upon the work of the second system. It addresses the issues with the natural language generation and properly expressing the reasoning behind the main characters actions and motivations. A background generation algorithm was added in fAIble III to provide a background history for the main character to give reasons for that character's emotions and choices in the story. This algorithm abstracted the main character's life into three stages: youth, teenager and adult. From these life stages events are picked stochastically to generate the character's background history. These events use story components from the story world and provide reasons behind the connections that the main character has with these story components. The result is improved sentences: 'Chris valued a sword' (from fAIble II) to 'Chris valued a sword because it was a gift from his father'.

A sample story from the fAIble III system is shown below:

> *"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. Anthony got away from Martin into a dirt road. He decided to go on a quest to get it back from Anthony. Martin noticed Thomas. He scared Martin. He decided not to ask Thomas for help. Martin looked for a way to journey Wallow Forest. He saw a carriage. He boarded it. He went from a church to Wallow Forest. He wandered to a grotto. He arrived at the grotto. He found a herb. He picked it up. He got off it. He went from the grotto to a dirt road. He noticed Kate. Martin asked Kate, "Where is a war hammer?" She told Martin how to find it. He meandered to the grotto. He arrived at the grotto. He wandered to a sidewalk. He went from the sidewalk to a cathedral. He arrived at the cathedral. He met Anthony. Martin noticed Anthony. Martin swung at Anthony. He dodged Martin's attack. He swung at Anthony. Martin won the fight against Anthony. Although Martin was victorious, his adventure was not yet over. He cared about a war hammer because it was a gift. There was also Wallace, a princess. Martin disliked Wallace because they were friends but got in a fight. Wallace took Martin's war hammer. Wallace got away from Martin into a dirt road. He decided to go on a quest to get it back from Wallace. Martin went from the cathedral to a cobblestone path. He went from the cobblestone path to a seaport. He arrived at the seaport. He found a root. He picked it up. He found a fairy nectar. He picked it up. He went from the seaport to a cobblestone path. He meandered to a tavern. He arrived at the tavern. He found a herb. He got it. He found a chisel. He got it. He found a war hammer. He picked it up. He noticed Christine. Martin asked Christine, "Where is a war hammer?" She told Martin how to find it. He went from the tavern to a dirt road. He went from the dirt road to a seaport. He arrived at the seaport. He found a plow. He picked it up. He found a herb. He picked it up. He found a plow. He picked it up. He found Exalibur. He picked it up because it was better than his war hammer. He found a mace. He did not get it because it was worse than his Exalibur. He met Wallace. He apologized to Martin. Wallace gave it to Martin. He successfully got it back."* (Bottoni et al., 2020)

Note the explanations of some of the actions in the sample story above. Like Campfire, fAIble can produce reasonably long stories that provide opportunities for truly entertaining stories. Unlike Campfire, however, it does not rely nearly as much on pre-defined story templates. The sample story above has a definite plot. Whereas there is perhaps too much wandering around the story world by Martin without seemingly any reason other than to chase Anthony, it does provide opportunities for story enhancements, such as what could happen when Martin reaches the Grotto and could possibly find other obstacles. Such side stories would enrich a narrative, and while this does not show in the sample story above, fAIble claims to have the tools (i.e., *story hooks*) available for such side stories to be developed.

GPT-2

GPT-2 (Radford et al., 2019a) is an open-sourced AI project by OpenAI that can generate extended text given an initial text input. Traditionally, such systems have utilised supervised learning techniques, where the system is trained to mimic certain behaviours, such as image classification. However, this leads to an inability for the system to generalise and expand into different tasks. Unlike other narrative generation systems, GPT-2 is not explicitly trained to create stories. Instead, it is trained via unsupervised learning on corpus of texts drawn from 8 million web pages (Radford et al., 2019b). This allows the AI to work from a large and varied dataset that contains text in different contexts and applications. The AI then creates text by predicting the next word, one at a time, through language modelling. Language modelling is the use of probabilistic and statistical techniques to predict the probability of a sequence of words (Rouse & Lutkevich, 2020). In this way, the system can generate logical text given all the previous text it has been given and/or generated.

GPT-2 is a novel system that can generate the new text following the same style, theme, and ideas of the input text. However, because of concerns from the creators over possible nefarious uses of the fully trained system, only a smaller model has been publicly released.

This model is available for testing online and works by taking in text input, from as short as one word tor up to several paragraphs. To demonstrate the capacity of GPT-2, we fed GPT-2 some lines generated from the fAIble III excerpts, which are shown in bold. The rest of the text was generated by GPT-2.

> ***Once upon a time there was Martin, a prince*** *of a small coastal African country. Martin liked most of his subjects, who had a knack for unruly behavior when left to their own devices. Nevertheless, Martin's kingdom prospered. The people would travel to the city from miles around to shop at the finest vendors, then go to one of the castle's three courtyards to spend their coin on the finest food and drink. It was a nice way to live, even if Martin could only afford to give each villager one meal a week. But then Martin met his rival, …*

> ***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.*** *This rivalry continued, despite Martin marrying the object of Anthony's desires. During the wedding reception, Martin and Anthony turned to stare at each other. They stared, until Anthony broke his gaze and moved away. Martin chased after him, grabbed him, and tackled him to the floor. As the guests cheered, Martin sat on Anthony. With one hand gripping the jeweled handle of his war hammer, and the other holding Anthony's neck. "You think a spear is going to stop me, eh? You think this weapon …*

The excerpts above demonstrate the ability of GPT-2 to continue the story with the same characters and themes. It also demonstrates how given more text, and therefore more context, it can create a completely different story. Some limitations to GPT-2 as a narrative generation system are that since it is not trained to create stories, story text can be generated that will never reach a (satisfactory) end state. If left to generate lengthy texts, it can also begin to become nonsensical. Nevertheless, the stories it does generate are impressive, especially for a system not targeted for automated storytelling. The sentences are diverse and syntactically correct, which is an area in which many narrative generation systems can struggle. The GPT-2 system can perform other tasks such as reading comprehension, summarisation, translation, and question answering. Overall, we can conclude that this system highlights the importance of having large and sufficiently diverse training texts, as well as the potential of unsupervised learning over task-based learning.

C2PO

Causal, Commonsense Plot Ordering, or C2PO (Ammanabrolu et al., 2020), focuses on the use of soft-causal relations for story generation. Automated narratives are often built upon *hard-causal* relations that move the plot forward based on explicit action pre-conditions and post-conditions. In contrast, *soft-causal* relations rely on the reader's common sense to understand the relationships between events. These relations allow the reader to infer events that may have occurred in the past. For example, if a character dislikes another character, it may be inferred that they will want a weapon.

Narrative generation by C2PO is approached through plot-infilling, where a high-level plot outline is filled in by creating branching sets of events that connect the high-level plot points. The plot outline is generated by extracting high-level plot points from story text using a pre-trained neural coreference resolution model to identify coreference clusters. These clusters represent expressions in the text that refer to the same character or object. They also extract a triple from the story composed of a subject, relation and object, and align it with the coreference clusters for the chosen character. The result is then a temporally ordered sequence of events where the character serves as the protagonist of the triple. These triples are used to create the subject-relation-object phrase that describes the plot point.

The next step is to generate the plot graph that will connect the events between the plot points. This is done through the COMET (Bosselut et al., 2019) system, which produces possible next events for the story. For two given plot points, p1 and p2, the COMET system is queried starting from p1 to generate the events caused by a character want. It is then queried from p2 to generate the needs relations, or pre-conditions for the actions. The result is two directed acyclic graphs. From here, the system then looks for the optimal links between the plot points by computing the probabilities. This is then linked together to create the overall plot graph, which can be traversed to generate the story.

Below is a sample fairy tale story created by the system. The sentences in bold represent the initial high-level plot points used to generate the story.

> ***Queen asks her mirror.*** *Queen wants to look better. Queen wants to try on clothes. Queen starts to be mad.* ***Queen is furious***. *Queen tries to relax. Queen wants to take a nap. Queen starts to get up. Queen begins to approach someone.* ***She appears at a dwarfs'***. *Queen starts to surprise everyone. Queen starts to have a party. queen wants to have money. Queen tries to buy poison comb.* ***She brushes with poisoned comb***. *Queen tries to wash her hair. Queen starts dry it. Queen wants to be hungry. Queen wants to get the knife.* ***Queen cuts the apple in half.*** (Ammanabrolu et al., 2020)

This story highlights the use of soft-causal relations for generating the story. Rather than explicitly stating all events that must occur, the reader can infer them. For example, with the sentences 'Queen wants to get the knife. Queen cuts the apple in half.', it can be reasoned that the Queen acquired the knife to cut the apple. While the sentences are still rudimentary, the system demonstrates a different approach to story generation that does not rely on explicit relations between events. Not everything needs to be explained in stories; this can allow the readers to fill in the story with their own imagination.

Character-Centric Neural Model

The work by Liu et al. (2020) investigates how a character-centric neural model can improve the consistency and explainability of stories. This is important, as neural models can often run into issues with keeping consistent plots, themes, and characters in the resulting narratives because of the black-box nature of neural networks[16]. By adding character embeddings, Liu et al. propose that the stories will be more consistent, as the actions are guided by the characters and their attributes and as a result, those actions will be explainable. Their neural model works by considering three main elements: character, situation, and action. The story progresses by having the model determine a character action based on their current situation. This is then generated into a complete sentence and the previous action is used as input for the next, such that the story is generated one sentence at a time. This is done to allow a fine control of the story plot.

The model is composed of an action predictor and a sentence generator. The action predictor uses a one-layer bidirectional LSTM network that generates the context embedding at each time step, based on the previous sentence and context embedding. With the resulting context embedding (or current situation), the character action will be predicted based on the character information.

This information is represented in the model as a character embedding, which contains relevant linguistic features (such as verbs and adjectives) as well as information on the gender, age, and personality of the character. Training for the model is done with a corpus of movie summary plots. Once training is complete, the action predictor can infer an action given a character embedding and context embedding.

The sentence generator uses two one-layer bidirectional LSTM networks to generate the natural language sentence representing the story event, one for encoding and one for decoding. The encoder is used to encode the context embedding and action, while the decoder is used to decode it into a sentence based on the character embedding. This allows the sentences to include information about the character, making the sentences more consistent throughout the story. For example, if the character is a basketball player, then when the action is 'play', the sentence will be about the character playing basketball. The character embedding information is also used to keep the sentences appropriate for the gender, age, and personality of the character.

Below is a sample story generated by the framework:

> *The plot focuses on a famous actor, Micheal. He hires a actor girl. When Micheal marries the actor girl, his wealthy parents leave him. Wendy tells him must deal with the problem, he decides not to talk. He leaves the town for his own. He tries to commit suicide. He swims through the river of town and drives some kilometers down the streets lines. He has been spending the night out. As the day progresses, Micheal finally divorces with her*. (Liu et al., 2020)

From this story sample, we can gain knowledge about the characters and their personalities based on the actions they take. Though the natural language is not perfect, it is understandable and consistent.

MABLE: MexicA's BaLlad machinE

MABLE (Singh et al., 2017), short for MexicA's BaLlad machinE, is a system which generates ballads based on the stories generated by the MEXICA system (Pérez y Pérez & Sharples, 2001). This system differs from narrative generation systems as it does not focus on creating a story plot but instead transforms one into a lyrical ballad. As the MABLE system is not burdened with plot creation, its focus is on incorporating rhyming and rhythm with sentimental themes to the story. The resulting ballad is made up of a series of paired lines, one of which progresses the narrative and one which is a poetic, figurative line that rhymes with the narrative one.

The system is composed of three main modules: a sentence evaluator, a sentiment analyser, and an integrator. The sentence evaluator module is fed a narrative line and creates a batch of candidate lines using parameters for the rhyme scheme, type and quality and the number of syllables. The candidate lines are generated using a Markov Model which is trained on a corpus of love songs. Candidate lines must rhyme with the last word of the narrative line, so a bag of words is used to provide suitable rhyming matches. To further expand the candidate lines, a second pass through the model is used where rhyming is based words that rhyme with any word in the bag of words. This also has the additional benefit of creating imperfect rhymes which are more realistic to how human written songs are. If no valid candidates are found, then a simple line of common interjections (e.g., 'oh oh') with a matching number of syllables is returned.

The sentiment analyser module then selects one of the candidate lines based on the total sentence sentiment. The sentiment score of a sentence is calculated by aggregating the individual sentiment scores of each word, with the end score indicating whether the sentence is positive, negative, or neutral. The candidate lines are then reduced to only include those which have the same overall sentiment as the narrative input line. From these, a random line is selected and sent to the integrator module. The integrator module is used to connect the narrative line with the generated figurative line by transforming the figurative line to third person perspective. This is needed as MEXICA generates its stories in third person perspective, and most of the corpus song lyrics used to train the Markov model are in first person.

Below, is a sample ballad from the MABLE system:

But she fell in love with him
Girl when they feel the same
The princess was in love with the priest
Can't let go and it never goes out
She also abominated what he did
Be the things they said
The princess was shocked by the priest's actions
And though her heart cant take it all happens

The excerpt above demonstrates MABLE's ability to create lyrical narratives. While the rhyming is not perfect, it clearly follows a rhyming scheme and is a great example of how automated narrative systems can build upon each other, much like how humans are inspired by other works.

The system was evaluated with human subjects against two other similar systems and human written lyrics. Rating was done based on coherence, emotional engagement, and overall lyric quality. MABLE ballads scored highest among the computer-generated lyrics for coherence and second for emotional engagement. The system creators note that this is likely a result of the wording choice of the highest-ranking system in emotional engagement and that MABLE could be improved by choosing more words that evoke higher emotional responses. Subjects preferred by far, the human written lyrics, in all categories. This was to be expected.

Trends in Non-Interactive Automated Narrative Generation Systems

Plot-based systems have tended to dominate the research works reviewed above. This makes sense as after all, stories are nothing if not a sequence of events that become interesting in the context of the previous events that have taken place. Yes, of course, the plot is far from being the entire story (so to speak) and character development and world creation play major roles in making a story enjoyable to read (or to listen), but few will argue that a story with a flawed plot can be made interesting only through good character development and/or creation of a rich story world.

Clearly, the most salient trend in these plot-based systems is the use of AI planning (and several variations thereof) as the basis of the processes employed to create interesting, logical and believable plots. So, while plot-planning is a clear trend among non-interactive systems, each system puts its own unique spin on the way they generate interesting narratives. Some systems look into incorporating literary devices to create more compelling stories. The IPOCL system withholds information to make the reader think and infer. The Provant and CPOCL systems introduce conflict and Dramatis and Suspenser seek to invoke suspense to create dramatic stories. HEADSPACE investigates how failure can increase character development. Flux Capacitor implements transformational arcs for characters, although it only results in a high-level story pitch. The C2PO system, which uses plot-infilling based on a high-level outline, could perhaps work well with the Flux Capacitor system. Lastly, a character-centric neural model uses previously opaque neural networks and succeeds in explaining the actions taken.

Re-usability is also a theme that is explored by the VB-POCL system and the work by Coman and Muñoz Avila. VB-POCL does so through story vignettes, or fragments, while the latter uses Case-based Reasoning to re-purpose events.

Mexica-Impro makes use of multiple planning agents to create stories that no single agent could generate alone.

There are some exceptions to the use of (modified) traditional AI planners. One such major exception is systems that reuse story plans to increase diversity using Case-based Reasoning as the way to select the next event in the plot. The Campfire system selects contexts rather than events per se, and uses criteria such as which action in the most appropriate context, best progresses the story towards an objective. Its CCM approach is different from the mainstream planning approaches.

Also exceptions are systems that employ stochastic approaches to selecting the next events, such as AESOP and fAIble. These also use context to assist in selecting a contextually-appropriate next event, but it is an auxiliary process for them, unlike for Campfire, which employs a context-centric process.

Campfire is the only system reviewed that uses a lifelike avatar to orally narrate a story to the reader/listener. The avatar is not capable of communicating with the user in both directions, but this is a relatively easy feature to implement as a next step. While fAIble also attempts to use such a storyteller avatar, it is not of sufficiently good quality (yet) to qualify as such.

Lastly, Table 1 summarises the non-interactive systems reviewed in this paper and associates certain qualities to them. This table can serve as a concise summary of the reviews contained in this section. The second column states the type of plot planning technology employed by the system, while the third column offers general short notes on their applications, research objective of the system and/or salient features of the system.

Interactive Narrative Generation

In interactive narrative generation, the user can participate in the narrative generation process and/or in the execution of the plot as the story is told. Thus, he/she can directly affect the outcome of the story – in sometimes unpredictable ways. This can be done at distinct levels of user involvement through a variety of different methods. A common method used is to have the listener take on the role of a character (e.g., the protagonist), wherein s/he can make decisions on what happens next – the actions of the character. A user can also act as an observer of a story world, which s/he can manipulate (e.g., trigger a storm). Whether a user needs to participate or not also varies. Some systems require the user to continue making decisions throughout the story (such as in video games) in order to reach a conclusion, while others merely allow, but do not require such interaction (e.g., Campfire).

Table 1. Summary of Non-interactive Systems Reviewed.

System Name	Plot Creation Method	Comments
TALE-SPIN	Rule-based system	Creates moral stories, as in Aesop's Fables
Ani	Rules, knowledge & constraints	Graphic display of Cinderella-like story.
UNIVERSE	Pieces together Plot fragments	Character-based never-ending 'soap operas'
MINSTREL	Case-based Reasoning	Created stories about King Arthur using TRAM
Bailey (1999)	Heuristic search of story segments	Pieces together story segments from huer. search
BRUTUS	Rule-based system	Cross domain to create betrayal & deception
Mexica	Engagement & reflection cycles	Stories about ancient Mexican civilisations
STORYBOOK	Finite state narrative planner	Focus on NLG on pre-authored stories
MAKEBELIEVE	Fuzzy inferences based on causal chains	Requires an input sentence to start the process.
Social Planner	Rule-based system	Changing characters' mental state is basis for plot
IPOCL	Intentional Partial Order Planner	Algorithm to increase character believability
VB_POCL	Modified Partial Order Planner	Searches for stories to find best vignette for details
CPOCL	Modified Partial Order Planner	Injects conflict into the plot
Niehaus/Young	Modified Partial-order planner	Creates plots with intentional gaps for user inferences
Coman/Muñoz	Case-based Reasoning	Goal is to increase diversity in stories generated
Dramatis	Modifies given sequence of events	Selects actions to introduce suspense in the plot
Flux Capacitor	Transformative arc generator	Describes evolution of characters across the story
Mexica-Impro	Engagement & reflection cycles	Uses two agents in collectively created stories.
HEADSPACE	State-space planning	Allows characters to fail to produce their evolution
Provant	Counter Planning	Antagonist seeks to obstruct protagonist goals
Suspenser	Hierarchical part order causal link planner	Selects actions to introduce suspense
IRIS	Partial order planner	Plan permits revision of character intentions
Campfire	Context-centric planning – CCM	Template-based progression. Can be interactive
AESOP	Stochastic selection of event	Produces a plot outline with only primitive NLG
fAIble	Stochastic selection of event	Full NG system with character dev. & world creation
GPT-2	Non-supervised machine learning. No plot	Seeks to extend input text from what was learned
C2PO	'Soft-causal' planning.	Assumes the ability of reader to make inferences
Liu et al.	Explainable neural networks	Uses neural nets whose actions can be explained
MABLE	Pre-created plot used as input	Creates ballads with rhyming and rhythm

Interactive narratives are also applied in serious settings such as in training simulations or for educational purposes. Training applications of interactive narratives allow a user to practice skills in a virtual setting, where the use of these skills has an effect on the world and thus, the story's progression. Automated generation of these narratives could introduce variability over several training simulations, something currently needed in that discipline. This is particularly important in simulations used for tactical training in military operations or in civilian emergency management training where building the narratives behind the training exercises currently requires extensive human effort.

In the following sub-section, we first present a brief overview of historically notable works in interactive narrative generation. We follow that with a more detailed discussion of more recent systems.

Historical Overview of Interactive Narrative Systems

One of the first interactive systems reported was for The Oz Project (Bates, 1992). It used a special architecture called Tok that supported interactive characters in a story micro-world. The Tok architecture is used to represent character minds that have goals, reactions, emotions, and the ability to talk and acknowledge the world around them. Human users could then interact with the characters through a text-based interface.

Next came MOE (Weyhrauch, 1997), which simulated a text-based murder mystery game. MOE generated game-search trees that were used to determine the possible game and user moves at any given point in the story.

Machado and Paiva (1999) created an unnamed story creation environment that used a graphical interface that children could use to select characters and scenarios for a story. Each child could either play as one of the characters or watch the resulting story based on the selected inputs that the child chose. In the latter version, the children are also able to edit the story as they saw fit.

The FAÇADE system (Mateas, 2002; Mateas & Stern, 2005) allowed users to play different scenarios of pre-written stories based on character relationships. Depending on the user's action, different story paths were played out. The stories generated by this system were limited and relied completely on the user interaction to progress.

KIDSTORY (Bayon et al., 2003) is composed of tools that allow children to create their own stories using sounds and drawings. The story can then be retold using a KidPad. This system focused less on the automated generation of stories and more on providing tools for user interactivity. This system is similar to that of Machado and Paiva (1999) in that it provides an interface for children to create narratives. The main difference is that the Machado and Paiva system is able to collaborate with the children.

LOGTELL (Ciarlini et al., 2005) took a logic-based approach to story creation by using predicate calculus to encode the rules that guide event generation. During the scene creation, users could modify the rules to alter the story as they saw fit. Knowledge of predicate calculus was required to interact with the system.

Moving forward, the goals of interactive narrative systems have been to increase the level of user interaction, extend the ways users can manipulate the narrative, and make the narrative generation more dynamic. The next section reviews research reported in interactive narrative creation since 2010.

Recent Interactive Narrative Creation Systems

Here we discuss recent systems in greater detail. Each sub-section contains one reviewed system.

Riu

Riu (Ontañón & Zhu, 2010) is an interactive narrative generation system that generates stories through analogies. As mentioned earlier in this paper, Riu employs Case-based Reasoning to select the appropriate analogy. This approach differs from many plot-based systems that use AI planning techniques and tend to be goal driven. Another highlight of Riu is that the narrative text for the stories is created in parallel with the content generation. In most story generation systems, the language generation is left as a post process to the content generation.

The created stories are about a robot named Ales which has lost its memories but gets them back progressively as the story progresses. The user acts as the main character (the robot) and as s/he explores the story world, the robot's recovering memories affect the actions that it takes. The system does this by finding analogies between events in the story world and the robot's lost-but-being-regained memories. The memories found in the case base can be adapted to the current problem to create diversity in the stories.

Riu contains a pre-authored main story that is used by the system as the general progression of the story. The role of this pre-authored story is to define the different *scenes* that the robot will go through. Each of the scenes contains different decisions that can be made but will ultimately be influenced by the player as well as by the progressively-regained memories. After each scene, the narrative continues to the next one.

An example text from a story generated by Riu is shown below:

> *"One day, Ales was walking in the street when he saw a cat in front of him. Ales hesitated for a second about what to do with the cat. Ales played a lot with the cat and was very happy. But the cat died, leaving Ales really sad."* (Ontañón & Zhu, 2010)

Note that in the above sample story, Ales first hesitates as a result of losing its memory. As the correct analogy is later selected when it starts to gain its memory back, Ales proceeds to play with the cat to achieve subsequent happiness, as decided by the human participant. Another decision point arises in the story, and in this case, the analogy selected causes the cat to die, making Ales unhappy. One disadvantage is that the story provides no reason for the cat's demise.

Each scene in Riu works as follows: First, the scene is presented to the user. Second, the current scene is used to retrieve a memory from Ales' lost memories, which is then presented to the player as well, and added to the 'retrieved memories set'. Third, Ales has a set of emotions that it 'likes' (e.g., happiness, love, etc.) and a set of emotions it does not like (e.g., sadness, hate, etc.). Then given the 'retrieved memories set', Ales tries to infer what would happen if the player chooses each of the different actions s/he can choose. Basically, Ales imagines the future by analogy from its memory. If the future Ales imagines for an action includes emotions it does not like, it will refuse to do it (e.g., it does not want to play with the cat, because it imagines it will die, like Ales' pet bird did before). If the negative emotions in the imagined future are very strong, Riu will even remove the possible choice from the player. Moreover, if the positive emotions are very strong, Ales will directly perform that action without even giving the player a chance to choose.

Through this mechanism, the story progresses without the need for a final state or goal to be specified. Instead, events are triggered as the robot encounters different situations in the story world that are analogous to the pre-authored memories. The pre-authored memories have to be composed and loaded into the system by a human author/user.

The scenes in Riu stories are composed of a computer-understandable description and a human-understandable description. The computer-understandable descriptions are stored as graphs, with nodes representing objects, characters, actions, relations and attributes present within a scene. These nodes are then linked to sentences, which make up the human-understandable description. This provides an easy way to move between the computer-generated story and its human-readable version.

The system chooses the appropriate memories by evaluating the *surface* and *structural* similarities between the pre-authored memories and the current scene. The surface similarity is calculated based on keywords associated with the scenes and is performed first. This narrows down the options that must be evaluated for structural similarity, which would otherwise be a computationally expensive task. Once the most similar analogy has been chosen, the story text is generated by taking the set of sentences associated with the source scene and those in the chosen memory scene and using substitutions to generate the story text. The structural similarity component Riu uses is based on the SME computational analogy system (Falkenhainer et al., 1989). SME takes as input two graphs, and generates an analogical mapping between them, as well as a 'structure mapping score' (which is higher if SME can find a strong analogy between the two graphs). Riu uses this structure mapping score as the structural similarity. SME is computationally expensive, as it uses a search-based approach, so Riu tries to minimise the number of times that SME is called, hence the use of the surface similarity.

This process results in interesting stories that are perhaps more creative than merely a sequence of actions chosen by a planner. However, the system is limited in that it requires pre-authored stories from which to create analogies. Furthermore, to create diverse and interesting stories necessitates a vast amount of 'memories' from which the system can pull analogies. This increases the authorial burden, something not desirable in automated generation systems. Nevertheless, Riu takes a refreshing approach in the creative process of content and sentence generation.

MINSTREL Remixed

MINSTREL Remixed (Tearse et al., 2010) is a reconstruction of the original MINSTREL system (Turner, 1991). MINSTREL was a pioneering system in automated non-interactive narrative generation that also used Case-based Reasoning to generate stories. However, working copies of MINSTREL were lost after its author's death and a description of the system only remained in a published paper. MINSTREL was reconstructed to gain a better understanding of the concepts in the original system, modernise the system to today's standards, and begin work on transforming it to an interactive narrative application.

Both the original and the reconstructed systems use a case-based retrieval and adaptation process to generate stories. MINSTREL Remixed represents stories as graphs and contains two major components. The first is a Transform Recall Adapt Method (TRAM) process that performs granular story modifications by recalling information from a story library and adapting it to a new one that is then merged into the story. TRAM was the central feature of Turner's model of creativity, which MINSTREL was designed to showcase. The second is the Author Level Planning (ALP) system that enforces constraints on a high level, specifying details to be added by calling on TRAM. For example, if the ALP wants to include a knight dying, it can call upon TRAM, which works to find a matching story in the library, such as one about an ogre being killed by a knight. TRAM then transforms the story from the library to be of a knight being killed by another knight and thus adapts it to fit into the new story.

A major change from the original system to the new one was in the way that a TRAM is selected. In MINSTREL, if no matching story was found in the library then a random one was chosen. This resulted in creative but sometimes illogical stories. To resolve this issue, MINSTREL Remixed employs a graph distance algorithm that minimises the distances to find the closest match. However, this caused the creativity of the stories to suffer when compared to the original system.

MINSTREL originally used Planning Advice Themes – story templates based on adages – to produce its stories. The new system migrates from using Planning Advice Themes to Generalised Story Templates, which are similar but allow for greater flexibility as they are not limited to adages. The system was also updated to allow for switching between choice algorithms, or TRAMs, and was constructed to easily allow for interaction with users, who can select which TRAM algorithms are selected (there are several variations available for the user to select), thereby giving the user more control over the stories generated. MINSTREL Remixed also has the ability to go back to a point in the story and continue down a different path or alternate storyline. These changes pave the way for MINSTREL Remixed to become a fully interactive narrative system.

The MINSTREL Remixed system is important because it provides a working model of the original MISNTREL – a landmark storytelling system. The TRAM and ALP components provide an ingenious way to make the most of the knowledge base with which the system is provided a-priori. By re-using and adapting events, stories can be created that make sense and can involve different characters and settings. An interesting result of their research of the MINSTREL system was the loss of creativity in stories after the implementation of the graph distance algorithm was made to avoid illogical stories. This demonstrates that automated narrative generation systems may not need (or want!) perfect reasoning to create good stories. Perhaps leaving irrational elements in storytelling systems is the key to automating creativity and the occasional illogical story is a worthwhile price to pay. After all, stories written by human authors often use suspension of belief in their own works.

Picture Books and Picture Books 2

Picture Books 2 (Ang et al., 2011) is an expansion of the Picture Books system by Hong et al. (2009). Picture Books created stories for young children given a single scene input picture composed of the story elements (characters, objects, and backgrounds) selected by the child. This limited the narratives generated, as stories are typically made up of more than one scene or event. Picture Books 2 allows for children to input multiple scenes for a story, with a minimum scene requirement of three. They use a theme-based cause-effect planning algorithm to generate stories with moral lessons that enable children to explore the world within their picture-narrative environment.

The system is composed of four modules: the story editor, story planner, sentence planner, and story generator. The story editor is the interface used by the child to create the input picture scenes for the system. The story planner then produces a story plan based on binary semantic relationships of the story ontology. These relationships emulate the semantic network knowledge representation of ConceptNet (Liu & Singh, 2004). Unfortunately, the ontology and semantic relationships must be manually populated, which increases the authorial burden. A theme is chosen based on the main character's traits and the objects present in the input pictures. Through the semantic relationships and theme, a causal chain-of-events path is produced that is contextually relevant. Variety in the stories is handled by randomly selecting nodes to be included in the path. The sentence planner then lexicalises the story goal and concepts to send a set of sentence specifications to the story generator. The story generator uses simpleNLG (Venour & Reiter, 2008), a widely available tool, to convert the sentence specifications into natural language.

An excerpt of the realised text of a story by Picture Books 2 is shown below:

"[1] It was a fine evening.
[2] Danny the dog was in the camp for a trip.
[3] He buys a packed marshmallow.
[4] The camp is very big.
. . .
[5] He sees a shadow.
[6] Danny the dog feels scared.
[7] He does not know what to do.
. . .
[8] Since then, He learns to be brave." (Ang et al., 2011)

From the above sample story, we can see that although the statements are short, the events described actually make sense and follow on the prior ones. While not particularly entertaining for adults, it does a good job of creating stories for children, who are its target audience.

The Picture Books 2 system provides a user-friendly interface for its target population – young children. This is important, as other interactive narrative systems require a higher reading knowledge level from its users. Therefore, this system provides storytelling for a niche group which (surprisingly!) isn't commonly targeted by the average automated narrative generation system. While the grammar of the story output is not perfect, it is competent at keeping sentences simple for its audience. However, during testing it was found that the population of the semantic ontology had to be done carefully to avoid illogical story paths. Cautious planning of the semantic ontology resolves this issue, though. Overall, the Picture Books 2 system takes a novel approach to automated storytelling for its niche target audience.

Brain Computer Interface for Narrative Generation

Gilroy et al. (2013) took a novel approach to user interaction in their interactive narrative system by incorporating EEG readings as the media for their user interaction. In their unnamed system, instead of focusing on a plot-centric narrative, they shift towards a focus on a user's affective responses to characters. The enjoyment of stories is linked to having relatable protagonists, where the bond between the reader and the main character is strong. This empathetic support can be measured through EEG readings, which measure electrical brain activity through neurofeedback of frontal asymmetry. Higher relative left frontal activity in the brain indicates a greater empathy felt towards the protagonist. These readings can then be utilised to shape a narrative, leading to a deeper interaction between the user and the story generation process.

This system is an extension of their previous interactive storytelling system. In the original system, a planning approach was implemented for generating a sequence of narrative events, which were then staged in a 3D game engine. The stories created revolve around a female junior doctor who faces adverse events throughout the narrative. The visuals of the 3D rendering and filmic shooting conventions are used to induce empathic feelings towards the protagonist. The main addition to the system is a Brain-Computer Interface (BCI) that allows the user to interact with the story generation in real time.

The BCI takes in neurofeedback denoting feelings of empathy or support from the user. As the protagonist goes through difficult situations, the user is prompted to support the main character. The outcome of the story is dependent upon the final level of support demonstrated by the user. The system uses landmarks, or representations of narrative situations, to include appropriate dramatic content and enable narrative control and progression. They are chosen from a set of possible options and selected non-deterministically to increase story diversity. Depending on the user's measured support, different narrative actions are chosen. The greater the support a user shows through the BCI, the more positive the outcome. On the other hand, if the user fails to give support, then the protagonist's situation worsens. Neurofeedback visuals are used to show the users how their feedback is being interpreted so that they can adjust their input accordingly.

The BCI introduces a novel way for the user to interact with the narrative by involving the user at a biological response level. In their studies, the researchers found that users were able to quickly learn how to use the BCI. Still, the stories output by the system are limited to a finite set of outputs dependent on the amount of user interaction. Combining biological user interaction with dynamic event generation could be the next step for these types of systems.

IRIS Interactive

The IRIS system was originally a non-interactive system that incorporated intention revision in its characters, as reviewed in 3.2.12 above. The original (non-interactive) IRIS system was converted to an interactive system in Fendt and Young (2014), who studied the transition from non-interactive narrative automation to an interactive narrative system for a video game. The IRIS narrative generation system created story outlines that had localised suspenseful moments. The adaptation changes the IRIS system into a text-based adventure game that allows for user interaction while still maintaining elements of intention revision within the narrative. Here we only review the interactive version – refer to section 3.2.12 above for a review of the original, non-interactive version. We found it interesting that in the original version of IRIS as described in (Fendt & Young, 2011), no mention is made of the goal to create suspense in the story. In fact, that paper states that the objective of IRIS is to incorporate intention revision in the characters to make them more interesting. Yet, the paper describing the interactive version (Fendt & Young, 2014) cites their earlier paper and states that the objective of the original IRIS was to create suspense: '*The IRIS narrative generation system (*Fendt & Young, 2011*) is a tool used to generate story outlines with built-in suspense . . .* ' (Fendt & Young, 2014, p. 244), but no mention of intentionality revision. We found this to be confusing.

The text-based narrative game developed is set in a western heist domain and makes the player a bank robber. The player can then interact with the story world by making decisions (e.g., picking up items, talking with other characters in the story) based on a set of choices provided to the player by the system. The main game goal is for the player to rob the bank and get away with the loot. The story is then broken down into sub-goals, some being optional while others mandatory. There are three mandatory sub-goals involved with suspense in the narrative: using dynamite to blow up the vault door, using a key to open the safe, and moving the loot to the train. The emphasis of this game is to elicit suspense in the test subject as he/she plays the game.

When modifying the system to become interactive, two major concepts were addressed. Shifting part of the narrative control to the player can make it difficult to maintain the story plan and the suspense. If a player is unable to find the solution or path to the game goals, they may become frustrated and quit. The player should also not be able to bypass the suspense action sequences because then the suspenseful quality of the game will be lost. This is handled by the system through two transformation mechanisms. In-game hints are used to help guide the player through the story and to not allow her/him to stray from the game objectives. The narrative control between the user and the system is balanced by reducing the player agency during the key suspenseful moments, allowing the scene to play out without being interrupted or avoided. This is done by providing the user with a dynamic list of possible actions that change depending on the story state.

During testing, they found that the use of the two transformations combined resulted in more suspenseful game experiences than when one or none were used. The approaches taken in the development of this interactive system show ways that non-interactive narrative systems can be converted to interactive systems as well as how suspense can be maintained, even with the loss of some narrative control. The IRIS adaptation is limited in its story setting and narrative. This is probably because the creation of suspense in stories is difficult to automate. Future research could investigate other ways of allowing players to interact with the game world and the adaptation of the story into different domains.

Aesop

Aesop (Meo et al., 2018) (not to be confused with the previously reviewed AESOP) takes a unique approach for the user's interaction with the system. Unlike other interactive systems, Aesop utilises both verbal and nonverbal communication from the users. The system's goal is to allow for the collaboration of an AI agent and human user to work as equals in a creative process. This is also different from other narrative generation systems where the user provides directions to an AI agent who then carries them out. Aesop is more of a tool to study how to interact with an AI agent in the form of story generator than it is a narrative generation system per se. Its authors refer to it as a storytelling system, rather than a narrative generator.

The Aesop system is composed of four main components: a language parser, a gesture tracking model, a composition graph module, and a dialogue manager. The language parser and gesture tracking model are used to receive input from the user. They use the language parser, TRIPS, which employs a language ontology amplified with information from the visual domain. User gestures are tracked using a Multimodal Behaviour Analytics system that uses a Microsoft Kinect sensor. The combination of information gathered from the language parser and gesture tracker generate a semantic representation of events, entities, and relationships through a directed acyclic graph (a composition graph).

The resulting composition graph is used to reason about relationships and serves as an intermediate representation between the visual domain and the language domain. These graphs can be expanded by the users through the language parser and gesture tracking components of the system. The dialog manager uses the composition graph to create animations using Muvizu animation software that are then presented to the user. The dialogue manager is also responsible for handling the collaborative exchange between the AI agent and the user. For example, if the system finds that the composition graph lacks information, then the dialogue manager will prompt the user for clarification.

Overall, the Aesop system uses a unique method for involving the user in the creative process of storytelling. It employs a more immersive interaction process by allowing users to interact verbally and through their body gestures, which allows users to interact with the story telling process in a unique way.

Campfire Interactive

This is the exact same system described and reviewed in section 3.2.13 above – we only describe here its interactive features. Whereas Campfire is fully able to operate non-interactively, it also allows for the possibility of dynamic story generation. In fact, one of its most interesting features is its ability to have a listener interrupt the story at any time as it is being told and make changes that will apply to the rest of the story. These changes to the story specification are then used to re-create the rest of the story (from the point it was interrupted) to possibly change its ultimate ending or change its genre. The system can also sometimes make the changes to the story at a point chronologically prior to the time of the interruption in order to make the story logical in cases where implementing the changes at the exact time of interruption would make the story illogical in some way. Because of this, this system could have easily been reviewed here as an interactive system. However, the user cannot be a player in the story, nor can he/she interact with the characters in any way in real time.

The most common approach for interactive storytelling has been to have pre-determined paths that allow for alternate endings based on the reader's choices at pre-determined decision points throughout the story. However, this requires the pre-authoring of each alternate ending and limits the stories to a finite, pre-defined set. The most impressive thing is that Campfire does this without relying on pre-determined decision points and alternative pre-planned paths by following story outlines as a guide. This outline can be carried across multiple genres – western, medieval or space odyssey adventures. This story outline is then re-generated by its CCM planner, and the details are subsequently filled in by the system. During extensive testing, it was found that the system was successful in dynamically changing the story based on user input.

Crystal Island – Narrative Adaptation

Crystal Island (J. Rowe et al., 2009) is a middle school learning environment for the study of biology that includes elements of simulation systems and intelligent tutoring systems. Within the Crystal Island environment, the students can explore the simulated world, interact with other characters, and create/test hypothesis. Narratives in Crystal Island are in fact lesson plans, carefully designed to educate users on particular subjects. Different narrative routes provide learning progressions tailored to the individual user. J.P. Rowe et al. (2010) investigated the idea of adapting narratives in Crystal Island to each student in order to achieve personally-tailored learning experiences. Their work is an example of narrative adaptation rather than narrative generation. It adapts pre-authored narratives to permit user agency as the user, through an agent that s/he controls, can explore the simulated world in the narrative. This is done primarily to tailor the learning experience for the user through the agent they use to explore the simulated world. Tailored user experiences can be particularly useful for educational purposes, where people learn at different rates and through different methods. However, there must be a balance between user agency and maintaining the narrative direction. Users should be able to direct their learning experience without deviating from the educational material.

J.P. Rowe et al. (2010) investigated several techniques to adapt narratives into interactive experiences specific to learning environments. The authors created a generalised framework for narrative adaptations for an interactive narrative director agent. The framework generated is composed of three parts: plot adaptation, discourse adaptation, and user tailoring. In plot adaptation, there is a structural manipulation of the sequence of events composing the plot. This is triggered by user actions or by the system to tailor the narrative to the user. It can help users who are struggling or bored by adding or removing goals to target weaknesses in the user's knowledge. There are two types of plot adaptations: direct and indirect. With direct plot adaptation, an explicit alteration of the plot trajectory is done by modifying major plot events. This must be done carefully so as not to cause conflicts with the existing narrative history. Indirect plot adaptation avoids the 'manipulated' feeling of direct plot adaptation by altering key narrative features that discreetly result in shaping the path of the narrative. This is accomplished by modifying virtual character states, introducing or removing virtual characters, adding rewards and incentives, or changing player abilities.

Discourse adaptation is important for defining the presentation of the learning material. Alterations can highlight moods, facets, and story elements relevant to the lessons. These changes can create a more enjoyable learning experience and serve as motivational factors. The discourse adaptations are done by modifying the perspective of the graphical environment through camera and lighting adjustments or through manipulating the order of events to conceal and reveal information. Examples of changing event order include foreshadowing and flashbacks. This can be helpful for providing hints to keep users on track with the lesson.

User tailoring is a key aspect of story-based learning environments. The right amount of support needs to be provided while still allowing for user agency and plot progression. This is done through narrative scaffolding, revealing information, and providing feedback through the narrative. Narrative scaffolding aids users by prompting them towards goals that lead to plot progression and by providing dynamic challenge levels for effective engagement and learning. The dynamic revelation of information also works to give a user the right amount of help. Meta-narrative feedback is provided by encouraging users to reflect on their choices and through explicit guiding of actions.

The framework provides good guidelines for adapting narratives in educational settings. Unlike in entertainment-oriented storytelling, special care must be taken to ensure that the underlying lesson of the story is properly delivered. The techniques discussed for adapting the narrative work together to create a unique experience for the user while preserving narrative coherence. This framework was the first of many research works aimed at enhancing education-focused narrative environments. As such, it was not intended to be comprehensive in its use of adaptation techniques. Rather, its purpose was intended to begin research into the types of technique that could be used in educational systems like Crystal Island. Their latest work has been on the personalisation of narrative generation with deep reinforcement learning techniques (Wang et al., 2017) (see section 4.2.9 immediately below).

Un-named System by Wang *et al.*

Wang et al. (2017) explore the use of deep reinforcement learning framework to improve interactive planners that personalise the narrative to the user. Their framework was used on the Crystal Island system (Rowe et al., 2014), which uses an open-world game environment with interactive narratives focused on education. Wang et al. sought to improve the planner that adapts the narrative to increase its effectiveness. The framework uses the normalised learning gain (NLG, not to be confused with natural language generation) metric, which is the normalised difference between the player's pre-test and post-test taken before and after playing through Crystal Island.

The framework has four tiers: a dataset tier, a player-simulation tier, a reinforcement learning (RL) tier, and a narrative adaptation tier. The dataset tier contains data from a player interaction corpus from Crystal Island generated through human studies. The data contains information about the player (such as gender and their NLG scores) and the actions taken in a play-through. These data are divided into a testing set made up of 80% of the original data and a test set from the remaining 20%. This training dataset is passed to the player-simulation tier.

The player-simulation tier is composed of a predictive bipartite model, with one module predicting a player's next action and the other module predicting the player's learning outcomes. This is done to simulate player behaviour with an interactive narrative planner. The action simulation is based on the player's interaction history and the possible player actions. Crystal Island contains 15 discrete player actions, with a subset of these triggering adaptations in the narrative. The prediction for the actions is then calculated through a 15-class classification problem which takes the user action history and player information to output a probability distribution for possible actions. Similarly, the outcome prediction is done through a 2-class classification task (based on high NLG and low NLG) with the same input as the action simulator. Both modules use a recurrent neural network LSTM that is trained on the training dataset passed to the player-simulation tier. The predictions are then sent to the RL tier, which uses a Q-network (a neural network implementation based of the Q-learning algorithm[17]) to optimise the planning policy for the interactive narrative planner. The improved planning policy is then evaluated in the narrative adaptation tier by applying it to the simulated players in the test set.

The framework was evaluated against a linear RL-based planner and was found to have significant performance increases with the Q-network planner. This was attributed to the framework's ability to identify non-linear player interaction patterns. Their results showed that the updated policy resulted in simulated players achieving a high NLG, representative of an increased learning gain. While this framework does not primarily focus on narrative generation, it provides insight on ways that deep learning can be used to improve planning policies, specifically for interactive narrative generation systems.

Plan, Write and Revise

We should bring attention to an innovative approach to story generation that has garnered some interest of late – that of collaborative story generation between human author and machine. The Plan, Write and Revise system (Goldfarb-Tarrant et al., 2019) offers the human author different levels of involvement (one being none at all). It is one of the objectives of the authors to determine what levels of collaboration work best to optimise the created narrative. It employs story planning using a method reported by Yao et al. (2019) that makes use of LSTMs. The system introduces multiple modes of operation. One of these is the cross-model mode that requires minimal interaction with the user, where the user enters a one-line topic and, optionally, a diversity value indicative of how much diversity the user specifies in the story. These are then sent to the *story planner* that generates a fabula (a term not used in their paper), and then to a *story writer* that generates the full story. The intra-model mode, on the other hand, involves much more interaction and allows the user to progressively participate in the writing of the story.

Un-named System by Porteous, Cavazza and Charles

While planning has been the dominant approach for plot generation, but there are certain problems associated with it. One of these is how does one shape the narrative effectively while maintaining real-time performance. This issue is present in non-interactive narrative generation systems, but becomes much more important in interactive system because when a user invokes modifications to stories constructed with planning – while the story is being told – this can call for plan adjustment or re-planning in real time, which can be computationally expensive. One way to tackle this issue is through a constraint-based planning algorithm (our label, not theirs) by Porteous et al. (2010). The idea behind this concept is that it is easier to plan for solutions that meet certain restrictions than it is to meet narrative preferences. This can also lead to surprising turns in the plot, something always desirable in fictional stories. While their work can be equally well applied to non-interactive systems, the authors specifically indicate that their work is in the context of interactive systems, so we place it under the interactive category.

In their algorithm, they approach planning through decomposition and by representing narrative control knowledge as constraints. Constraints are viewed as the key components of the plot structure in this algorithm, which represent desirable conditions that enrich the narrative. The constraints are chosen based on story world predicates, or on attributes of the main characters. They are used to describe the state of the story world and the condition of the characters, such as their current status with respect to the story state. These constraints can then be used by the planning system to shape the narrative into a desired story trajectory that will result in pleasing story outputs by restricting the search space based on the constraint category. The constraints are categorised into *hard* and *soft* constraints, where hard constraints must be strictly observed, and soft constraints are desirable but not necessary.

NetworkING

Porteous et al. (2013) make the argument that by enhancing the social relationships among the characters, the burden to generate detailed plots is reduced. This makes for greater diversity in the generated narratives when the relationships are modified. They argue (with supporting references to the literature) that authors typically initially think of stories in terms of characters, their inter-relationships and the situations in which they might find themselves. Plots, then, emerge naturally from those relationships.

Their Network for Interactive Narrative Generation system features a graphically-supported network that details all the relationships among the characters in the story to be generated. Each character has three classes of social interactions: affective, romantic, plus a default of indifference. A human user can interact with the system by modifying the relationships in the network through the network's user interface. The new context introduced by the new relationships can affect which actions the characters would choose to perform in the plot, leading to different outcomes. The system uses planning to generate the plots, but the relationships provide a rich set of conditions that interact with the pre-conditions of the actions to guide the direction of the resulting plots. The authors provide an example set in a medical drama, such as are common in television series of that genre.

Nothing for Dinner

Nothing for Dinner (Szilas et al., 2015) is an interactive computer game that allows users to interact with a story presented in 3D animation using the IDtension system (Szilas, 2003). The story places the user as a teenage boy whose father has suffered a cerebral accident, leaving him with hindered mental and behavioural capabilities. In the game, the users navigate making dinner while attempting to include their father, who proves to be difficult. If you cook without him, the father feels left out. If you try to get him to cook with you, he refuses, and if you do nothing, the father will try to cook but make a mess. During the playthrough, the user must make decisions, which then opens new possible choices, both for the player and the non-player characters (NPCs). This allows the story to go in different directions, making for a new narrative across playthroughs.

The IDtension system focuses on narrative properties – or narrative laws – to create its interactive narratives. The system breaks down the narrative into three layers: a discourse layer, a story layer, and a perception layer. The discourse layer represents the message or morality being conveyed by the narrative. The story layer represents the sequence of events and actions that compose the narrative. Lastly, the perception layer deals with how to make the narrative engaging to the readers so that they can understand and maintain interest in the story.

To create the narrative, IDtension uses five modules. A *story world* module that contains all story entities and states, a *narrative logic* module that calculates all possible actions for the characters given the story world contents, a *narrative sequencer* module that filters the actions from the narrative logic module, a *user model* module that contains the user state and estimates the impact of user actions, and finally, a *theatre* module that provides a display for the narrative and handles the interactions between the user and the system. To make the narrative interesting, the use of low and high-risk obstacles is used to create conflict. The user can then experience the narrative differently based on their knowledge of the obstacle. No knowledge leads to a surprise, while increased knowledge of a high-risk obstacle can lead to heightened suspense.

The system can then operate on two modes: automatic generation and first person. In automatic generation, the narrative sequencer picks the actions for the story. Alternatively, in the first-person mode, the user plays as a character and takes turns deciding on actions with the sequencer to generate the narrative.

Nothing for Dinner presents an immersive narrative generation system that allows the user to partake in a rich story world. However, as with any system, there are limitations. According to the review of one user (Short, 2016), s/he felt that at times her/his actions held little consequence in the story. Furthermore, conflicting actions could be performed consecutively and the other characters did not seem to maintain a memory of the previous actions. The reviewer also found that s/he could get stuck in loops consisting of the same actions, and because the non-playing characters do not seem to maintain a memory of the previous actions, nothing changed. The reviewer did note, however, that the story world was well developed and immersive, as the other characters interacting and conversing with each other could be observed without the need for user interaction.

AI Stories

AI Stories (Burtenshaw, 2020) is a proposed (the work does not appear to have been completed at the time of this writing) system for generating narratives cooperatively with children through conversation. The idea behind it is how language play in young children is used as part of their process of developing fundamental skills, and to express their interests. Burtenshaw suggests that while certain aspects of stories, such as character descriptions, need to remain static, interactive narratives can be fluid, allowing shifts in the story while maintaining an overall narrative path. This is especially important with children, who are curious and will ask questions, interrupting the narrative and shaping it at the same time.

The proposed AI Stories system takes in user input and responds using different subsystems that are selected to give the best response. While the number of subsystems can be expanded, only three are discussed: a topic-based system, a context-based system, and a poetry and humour system. The topic-based system will respond to user questions on predefined topics by searching through topic-specific text. The context-based system will respond to user input using a sequence-to-sequence neural network[18] for single-turn dialogue, where a user asks a question and the system responds. The poetry and humour system will generate humorous responses using templates. To avoid the dialogue from being repetitive, a selector system is used to decide which response is best via reinforcement learning for consistent conversations and lexical analysis. This system will also oversee assessing input and deciding whether a subsystem relates directly to it, such as if the user asks for a joke.

This system presents a novel way to generate stories interactively. Rather than focusing on traditional aspects of narratives, such as plot, characters, and conflict, it guides the story through dialogue with the user.

Terminal Time

The Terminal Time (Mateas et al., 2000) system generates historical interactive narratives that are presented to a group of viewers as a in a documentary film, making it a truly unique automated story generation experience. The films are generated based on one-thousand years of history and follows a ‘cookie-cutter’ documentary style which has a clear plot and portrays one prevailing interpretation of the historical events. It also makes use of imagery and audio to aid in presenting a specific narrative, which is influenced by viewers’ answers to polls on their internal biases and opinions throughout the film. This system focuses on creating narratives that challenge viewers' perceptions and beliefs by generating distorted narratives that exaggerate the answers from the polls.

The Terminal Time architecture is composed of a knowledge base, ideological goal trees, a rule-based natural language generator, rhetorical devices, and a database of indexed audio/visual elements. The knowledge base contains representations of historical events which provide the content for the documentary film. The goal trees represent the narrative bias through which the events are portrayed and are used to include and exclude events in the story. These goal trees are edited through the user input which essentially creates the parameters for the system. The natural language generator creates the narrative text to represent the historical events chosen through the goal trees. These events are then connected using rhetorical devices that are pre-authored connective texts. Once the full narrative is generated, each text section is linked to a multimedia element which best matches the text. Lastly, text-to-speech is used to finalise the film.

One of the unique features of the system is the way in which viewers experience the film and how their input is gathered. The documentaries are presented to viewers in a theatre, which adds a public and social element to the interactions. Throughout the film, viewers are asked three sets of questions which are used to shape the narrative. These questions are similar to marketing questionnaires and poll viewers on their opinions and biases. However, the way in which the input is gathered is quite public. The viewers must clap for their answers and their responses are gathered via a microphone in the theatre. The answer with the most applause is selected. This creates a competitive and collaborative input gathering experience, where viewers actively engage with each other.

The Terminal Time system differs from other narrative generation systems reviewed in that it creates a social viewing experience. It also uses historical events as the basis for its story contents, rather than fables or other fictional materials. This system takes an innovative approach to interactive narrative generation and is reminiscent of interactive films which came later, such as the Black Mirror: Bandersnatch film which came out in 2018.

FearNot!

FearNot! (Aylett et al., 2005; Vannini et al., 2011) is a virtual learning environment which seeks to address bullying through role-play style stories of social situations in which bullying occurs. The goal is to provide a safe learning environment that can enable bullying victims with coping skills and increase bystander empathy for the victims. FearNot! is intended to be used by children in primary school, as the simulations use cartoon-like characters which play different roles such as the victim, bully, and other bystanders which may be passive or engage in the bullying. The child views episodes of direct and indirect bullying and in between episodes partakes in the story by providing advice to the victim. This provides the child with a safe environment where they can learn from social scenarios that can be otherwise emotionally, if not physically, traumatising.

Based on Bratman's Belief, Desire and Intention theory (Bratman, 1992) (also see footnote 14), the FearNot! system creates its stories through emergent narratives, which are stories which have not been pre-authored or planned and instead emerge from the interactions between the user and the story agents. Only the story setup, high level plot, and character creation require any authorial activities from the system designers. The high-level plot is needed to guide the narrative along, but there is no pre-set ending or timeline for the story. Instead, the story events are generated from the agent actions. Story agents are set to perceive their environment and react to it. A partially ordered planner is used to select agent goals based on the most intense reaction that an agent is currently having. For example, if the bully agent insults the victim agent, the victim agent may feel angry and want to fight back. This becomes a new goal in the planner. However, all paths to achieving this goal have a chance of the victim agent getting hit back, and as the victim agent has been pre-set to be fearful, the agent will decrease the goal importance and do nothing. The victim agent may also cry, although this is not an explicit goal and rather a reaction to its environment. If the user interacts with the victim agent by giving advice to fight back, then the goal importance is increased and in the next episode they may fight back against the bully agent. This results in a narrative that is flexible and stems from both the interactions of the agents with their environment and with the user.

Testing was performed on the system in primary schools in the UK and in Germany. Children worked in the virtual environment for thirty minutes per week over three consecutive weeks. Results showed that children in the UK tended to have better coping skill than those in Germany, while German children showed increased coping strategy knowledge after the three weeks. Overall, they also showed that awareness on bullying was raised and empathy for bullying victims increased. Adding audio though a text-to-speech system was one of the areas of improvement that was identified with the system.

FearNot! is a clear example of how automated story generation can be used as a teaching tool for complex social realities. Its use of emergent narrative allows for a narrative that changes with user input, creating a story that involves the user in a meaningful way.

Computer Game-based Story World Generation in Support of Narratives

Interactive narratives offer a wide range of implementations. They are popular in stories and in adventure games, such as in *The Wolf Among Us*, where players can interact in a fairy tale story world as the big bad wolf (Willingham, 2013). Narratives in video games are an ideal candidate for interactive narrative automation. Video game narratives are commonly hard-coded by developers, and the narratives branch out depending on the player's actions in the game world. Automating the narrative creation process for video games could remove the authorial burden from the game developers while allowing for dynamic and responsive narratives that a user would enjoy.

There are some automated narrative generation works reported in the computer game discipline that could be considered interactive by virtue of their role in computer games, which by definition are interactive. The objectives of these systems are mainly to set up and manage the imaginary worlds where the characters must live and where the game is played. Thus, they generally address the third part of automated narrative generation systems – world creation and operation. They seek to automatically create a large imaginary environment within which the game can be played, with minimum development effort by the human user (player) in creating that world. We briefly discuss some of these in this section.

NPCAgency (Pickett et al., 2015) takes a very different direction from other interactive systems. It focuses on an element of stories that has been mostly overlooked in the literature: Non-Player Characters (NPCs). As we mentioned earlier, non-player characters are entities in the game worlds (the story worlds also) that play minor roles and are not controlled by the players. The NPCAgency system generates conversational NPCs that can be considered assets in a story because they help populate the story worlds. NPCAgency creates NPCs based on predefined specifications by the author or story developer about their imaginary worlds. The NPCs can have unique backstories and attributes, and can hold conversations about themselves with other characters. The availability of rich story worlds can result in improved narratives with relatively little effort on the part of the human authors.

Similar in some ways to NPCAgency, the work reported by Kelly et al. (2008) allows all NPCs to behave more dynamically than a manually-created script might permit. It uses a hierarchical task network (HTN) that takes abstract tasks and refines them into actions. The HTN encodes the video game world knowledge and uses planning to create objectives for all the NPCs, with a script for each NPC as output.

Caves of Qud (Grinblat & Bucklew, 2017) is a science-fantasy video game that generates unique biographies for the major characters (historical rulers) in its imaginary worlds. The narrative generation comes when the game engine creates a history for a ruler and automatically generates a description of that ruler's life and events in natural language for the human player to read. It does not generate a narrative per se.

Dwarf Fortress is also a science-fantasy video game. It is notable for the great lengths to which it goes to define the imaginary story worlds and describe the aspects of those worlds in minute detail. It is able to compose complete worlds automatically when a human player wishes to play the game. It does not generate a narrative either.

CONAN (Creation Of Novel Adventure Narrative) (Breault et al., 2021) is a Procedural Quest generation system to be used for quests given by NPCs in video games. Similar to NPCAgency, its purpose is to reduce the authorial burden on game developers while still providing a novel system that generates coherent results.

Review of Commercial Systems

In this section we review and discuss two commercial systems, Korsakow and Dramatica Pro to better put in perspective the differences between the research systems discussed in sections 3 and 4, and the very few commercial systems currently available in the market. Table 4 summarises their salient points.

Korsakow is a software tool that is used to create interactive films that can be displayed on a browser and provide viewers with a dynamic watching experience (Thalhofer, n.d.). The films are created using a series of small narrative units (SNUs). Users create each SNU via a user interface where they provide a media component (usually an audio, video, or image file), which is then tagged with 'in' and 'out' keywords. The 'in' keywords represent what the SNU is while the 'out' keywords represent how it connects to other SNUs. The Korsakow tool then creates a narrative by linking the SNUs together based on their relationships derived from the input keywords. The result is a dynamic film which viewers can navigate through in a rendered page in a browser. The resulting film does not follow any set paths or follow any linear timeline, which allows for unique viewing experiences.

Dramatica is a software tool developed to help writers structure their stories by analysing the story elements to ensure a cohesive narrative, or Storyform (Phillips, n.d.). The argument is that simple stories that don't create a complete Storyform, while entertaining, will not be memorable. The Dramatica software uses a patented story engine to create a meaningful structure view of the author's story, so that the author can identify the story direction and fill in any gaps.

The Dramatica tool takes in user input by asking the author questions on his/her story (both multiple choice and free response), to get an overarching view of the overall story and main character. Among these, twelve 'essential' questions are asked relating to main characters goals, growth, and approach as well as story limits, outcomes and the overall story concerns and goals. The questions themselves serve to help flesh out the story and identify holes or problematic areas. The user input is then passed through the Dramatica software and outputs reports on the structure of the characters, plot, theme, and genre. It is important to note, however, that Dramatica cannot write a story itself and relies solely on the user input for any meaningful output. The expectation is that the user will better understand her/his own story and how to write it by gaining insights through the reports from Dramatica.

Summarising and comparing these systems, Dramatica and Korsakow are examples of narrative generation on a commercial level. Both tools require licences that users need to purchase to use their systems. While they provide unique services, it is important to note that much of the authorial burden still lies with the user. The user is the one deriving key insights via the Dramatica tool to create their story, which Dramatica cannot do on its own. Korsakow relies on the user providing keywords which create meaningful connections between SNUs, otherwise the result may be a chaotic, though artistic, narrative.

Trends in Interactive Automated Narrative Generation Systems

We have reviewed several interactive narrative generation systems in this section – from the historically significant ones to the more recent.

Traditional AI planners were not as prevalent in the interactive systems as they were in the non-interactive ones. Of course, that is because the injection of the human influence can make it difficult to pre-plan the plot. Moreover, the systems reviewed didn't always produce a full, listener-ready story to be told. In several cases, the interaction was part of a computer game, which by definition, are interactive in nature. Nevertheless, several interesting approaches were reported. We summarise these next.

Riu and MINSTREL Remixed are similar in that both employ Case-based Reasoning to determine the next event in the story. Both rely on the similarities of the current situation in the story to cases stored in the case base (called *memories* in Riu). It is also interesting that the similarity and adaptation process of MINSTREL, when left to itself, produced creative, although sometimes illogical set of events. When a more deterministic technique was used to reduce the incidence of illogical events, the illogical events were indeed eliminated, but at the cost of noticeably diminished creativity.

With regards to how to handle the user interaction, Gilroy et al.'s work with the brain-computer interface is the most intriguing of all those reviewed. It allows the user to participate passively (mostly) by having the BCI measure her/his feelings directly from the brain through an EEG. While this may reduce the fun factor for most players, it would be very applicable for the disabled players who could not otherwise participate. Aesop also takes an innovative approach to effect the interaction by including gestures in addition to text.

Campfire was briefly reviewed here again to discuss its interactive features beyond what was reported above in section 3.2.13. It contains one very interesting additional feature – the (optional) ability for the user to change the story in mid-stream without using pre-determined paths. This can be done at any time during the telling of the story.

A few of the systems focused on helping children author stories.

Lastly, the four systems reviewed under video games share their focus on story world creation, including generating backstories for the characters involved. This is important in interactive narrative generation overall, but it is particularly important in video games where the enjoyment is nearly always in actively playing the game and not so much on the other fine aspects of a story – its plot and its characters – that are more important for stories that are to be listened or read, rather than acted or played.

Table 2 summarises the interactive narrative generation systems discussed above. The table provides an indication of the plot creation means used as well as a comment on the focus of the research. Table 3 summarises the systems reviewed that are not actually narrative generators, but story world creators used mainly in video games. Table 4 summarises the brief discussions of two commercially-available systems.

Recent Trends in Narrative Generation System Evaluation

One of the most basic tenets of research is to evaluate the results of one's work to assess its success in light of the hypothesis set forth for the work. Does it achieve what it was hypothesised to do? Does it represent an extension of the state of the art? If so, then by how much? A widely-accepted set of standard metrics with which all researchers can evaluate their systems is always desirable when seeking to compare newly-presented systems or techniques to the state of the art. Many research disciplines have been able to develop such metrics (e.g., algorithmic complexity, speed of execution, lines of code, memory usage, and many others). Unfortunately, automated narrative generation is not one of these disciplines. The main complicating factor is the human element that is inextricably involved in the story. It always comes down to this: does the reader/listener enjoy the story? Nevertheless, several of the works reviewed take interesting approaches to evaluate the 'goodness' and effectiveness of their narrative generation systems.

There are three main facets to the assessment of a narrative generation system: 1) does its output reflect what its developers designed it to do? We call this *functional testing* and it is generally done internally by the research team. 2) Does this output have the desired effect on the target audience? This is *human-based testing* and as such, requires human test subjects. It asks

Table 2. Summary of Interactive Systems Reviewed.

System Name	Plot Creation Method	Comments
The Oz Project	Tok creates interactive charact.	Humans can interact with the system
MOE	Game search tree	Simulates murder mystery game
Machado & Paiva	A story creation environment.	Produces graphical interface that children can use
Facade	Pre-written stories	Defined alternative paths for a reader to play
Kidstory	Children interaction	Kids create their own story with interface provided
LOGTELL	Predicate calculus encodes rules	User can modify these rules to alter story as desired
Riu	Case-based Reasoning	Full narrative generation through CBR
MINSTREL Remix	Case-based Reasoning	Reconstruction of original MINSTREL but interactive
Picture Books	Contextual cause-and-effect	For children authors – pictures to form story
Picture Books 2	Contextual cause-and-effect	Same except accepts sequence of pictures
Brain/Comp. I/F	Not relevant to this system	Determines feasibility of brain-system interface
IRIS Interactive	Partial Order Planning	Converts IRIS into suspenseful interactive story
Aesop	No planning technique reported	Allows non-verbal communication with system
Campfire Interact	Context-centric reasoning – CCM	Can work as interactive or non-interactive
Crystal Island	No planning technique reported	Adaptation of narrative to personalise learning
Wang et al.	Reinforcement Learning	Improve adaptation of a narrative for Crystal Island
Plan-Write-Revise	LSTM networks	Interaction between human & system authors
Porteous et al.	Constraint-based planning	Planning and re-planning in real time was the goal
NetworkING	Relationship among characters	Character-based syst. Plots influenced by relations
Nothing 4 Dinner	User makes decisions to choices	User navigates story world according to its narrative
AI Stories	Creates narrative cooperatively	. with children through a conversation
Terminal Time	Uses historical events and goal trees shaped by viewers	An audience interacts with the system to create a dynamic documentary
FearNot!	Emergent narrative with partial order planning	Virtual learning environment to raise bullying awareness through role-play

Table 3. Summary of Video-game World Creation Systems.

System Name	Element of Story World Created	Comment
NPCAgency	Generates conversational NPCs	. . . with backstories. Leads to rich story world.
Kelly et al.	Hierarchical Task Networks plan	Creates objectives for Non-playing characters
Caves of Qud	Creates characters biographies	Leads to enriched story world for video games.
Dwarf Fortress	Composes complete world	. . . in minute detail for video games. Very rich worlds
CONAN	Quest generation system	Seeks to reduce authorial burden.

Table 4. Summary of Commercial Systems.

System Name	Authorial Burden	Comment
Dramatica Pro	Requires users to answer questions on their story; Does not create a story by itself	Analyzes and provides reports on an author's story to highlight any problematic areas.
Korzakow	Requires users to provide the meaningful connections between their multimedia components to create a story	Creates interactive films composed of small narrative units which are dynamically linked together based on their relationships.

questions such as: do human test subjects enjoy the story created? What do they like about it? What do they dislike about it? Is the story easy to understand? Is the length appropriate? Is the user interface effective? Etc. 3) Are there any metrics that can be used to quantitatively measure some aspects of the stories? As it turns out, some such methods do exist, albeit limited in scope, and are discussed in this review.

The primary objective of this section is to review the evaluation procedures followed by the authors of the systems reviewed in this article, and not to report on whether the evaluations for each system were successful or not. However, information about the latter is briefly included for the reader's benefit. Not all of the works reviewed were formally assessed by their authors. Those not assessed are only implicitly identified by their absence in this discussion. We should note that several of the reviewed works provided examples of how their system works and proceeded to analyse the example. While this is valuable for demonstration and illustrative purposes, we do not consider this to be a formal evaluation, and therefore do not include them in this section.

Functional Testing

This process seeks to determine whether the system being evaluated does what its developers designed it to do, i.e., does it work as intended? The term 'functional testing' in the context of narrative generation systems was used in Hollister's dissertation (Hollister, 2016). He specifically sought to determine whether his Campfire system's feature of real time story modification actually modified the story correctly. While this process is normally done internally by the research team, it is interesting that he used a third party human arbiter (rather than himself) to avoid the possibility of developer bias.

In Hollister's method a set of stories was generated by the Campfire prototype. Pre-specified changes were entered into the system by a human operator (Hollister himself) at different points in the story. Campfire executed the requested changes and then described them in a detailed manner as part of the story itself. For example, if the main character was being converted from a Wizard into a Knight, Campfire would state that the Wizard will now be converted into a Knight. Fifty such tests (stories with different change requests) were performed by Hollister. The story's text output produced after implementation of the changes was assessed by a single unbiased test subject (a Ph.D. student in Computer Science) who served as the evaluator. The evaluator used his judgement to determine whether the pre-specified changes were implemented correctly by Campfire and whether the revised story correctly exhibited the changes requested. No attempt was made to judge the stories' entertainment value. Half of the 50 tests used the same initial story to make different changes while the other 25 tests used different initial stories. The evaluator determined that all the changes in all fifty tests were made correctly by Campfire. See (Hollister, 2016) for further details on this test. While an argument could be made that the presence of the human evaluator would make this human-based testing, the role of the evaluator was to determine whether the story implemented the changes correctly – a mostly objective decision. It was more like checking the answers given on a test against a master set of correct answers, and not measuring the human's qualitative opinions about the stories. Therefore, we deemed this to be a functional test.

Ontañón and Zhu (2010) also used functional testing in Riu, where the system generated three stories with different memories. The stories were modified by the memories retrieved from its case base, and the results were assessed by human evaluators (the authors themselves) to see if it had been done correctly. The authors verified that the stories were generated as designed, thus validating the system's performance, albeit on a small sample size.

The BCI system (Gilroy et al., 2013) evaluation was interesting in that it was evaluated for what we would call functional testing – to see whether the interface performed as designed/expected. However, because the nature of the functional tests was to assess the effectiveness of a direct brain-computer interface, human brains became necessary. So, human test subjects had to be involved. Fifteen test subjects were used and subjected to various narrative-based stimuli while on an EEG and MRI. The EEG and MRI scans were computationally analysed to draw out evidence of the desired effects. The analysis involved a rather complex process of scan interpretation. The authors determined that the results were reported as generally positive.

The CONAN system (Breault et al., 2021) was function-tested (the authors do not use this term) to determine the diversity of the quest stories generated by their system. The standard of comparison was human-authored quest stories for video games. The stories generated by CONAN are plans that reflect the sequence of events from current state to goal state. These stories are based on strategies underlying various motivations (nine motivations were used). To perform the evaluation, the authors used a built-in classifier that identified the strategies and the motivations behind the strategies inherent in each presented quest to determine whether all nine motivations were represented. If so, then the set of quests was deemed diverse and the test successful. The classifier itself was validated by comparing its output classifications for 50 quest stories to those of two of the authors personally classifying the same 50 stories. While the correlation among the humans and CONAN was only moderate, it was considered sufficiently good in light of the equally moderate agreement between the two human classifiers, and the inherent difficulty of the classification task. With the validated classifier, the evaluation consisted of generating several thousand (actual number not provided) quest stories in two story worlds – a large one called simply Large World, and a smaller one called modified Aladdin – and applying the classifier to identify the strategies and especially the motivation for each quest. The results showed that all nine motivations were represented.

The Provant system (Porteous & Lindsay, 2019) was function-tested to verify their hypothesis that the system could 'complexify' a traditional story by integrating the obstacle-based conflict approach espoused by the system. Six narratives were generated by Provant in six different domains (e.g., Aladdin, Little red riding hood, Prom week, western theme, detective theme and Raiders of the lost ark). The narratives were then computationally analysed to determine the length of the narrative with and without protagonist interference. It also counted instances where non-recoverable obstacles were introduced by the antagonist. They also performed a human-based test to complement their functional test. This is discussed in the section below.

The NetworkING system (Porteous et al., 2013) was function-tested to determine the diversity of narratives generated after social relationships among the characters were modified. Fifty random problem instances were generated consisting of initial state and goal, each with four versions consisting of different sub-goals. Then, four versions of each of the resulting 200 narratives were created by changing the relationships among the characters, making a total of 800 test narratives. The Levenshtein distance between narratives was computed on a pair-wise comparison of narratives that counted the differences between the two narratives being compared. The larger this distance was, the more different the stories were and therefore the more diverse they were.

Weyhrauch's (1997) Moe system was functionally tested with nine user models that simulated how nine different types of users would interact with Moe. The objective was to evaluate the effectiveness of three search algorithms developed as part of his research.

Human-based Evaluation

Human-based evaluation was common in the works reviewed that reported a formal evaluation. The most obvious way to determine what people think about something is to ask them! This is only natural. Self-reporting user surveys have been an essential research tool in education and the social sciences for many years. This equally applies to research in computer science that involves human judgement or opinion. Needless to say, many of the works reviewed employed user surveys to assess their work. The general modus operandi was to give the test subjects stories to read (or with which to interact in some way) and then ask them questions afterwards in a survey about what they thought. Likert-type responses were common, either three-point or five-point. The questions asked, however, were very system specific and revolved around what the authors were trying to assess about their systems.

The fAIble system evaluation serves for us here as a template for the most common types of human-based evaluations. The various research groups (there were four successive ones) recruited and surveyed human test subjects who read fAIble stories. Four independent rounds of evaluations were performed – one after each of the first three versions (fAIble I, II and III) were completed (one year apart from each other) and then one final one that was more extensive than the first three and on all three versions at the same time (plus AESOP, which is considered fAIble 0). The three individual surveys asked a core set of six questions that were common to the three versions of fAIble's, plus a handful of other questions that were specific to each version. There are also two open-ended response questions to obtain general free-form feedback from the test subjects. The six common questions were:

(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?

Additionally, as an example of questions specific to two of the versions, the following question was asked in the fAIble II and III surveys (neither AESOP or fAIble I had a graphic user interface).

(1) Does the animation enhance the experience of the story?

The three individual assessments employed 41, 75 and 61 test subjects respectively. These test subjects were presented with two stories generated by the respective prototype system and asked to respond with 2 if her/his response to the question was 'yes'; 1 if it was 'borderline yes'; and 0 if it was 'no'. The researchers expected to see meaningful improvement from one version to the next. This certainly was true between fAIble I and II, but not so between II and III. While a possible cause was the demographics of the test subjects (all were college students in the evaluation of fAIble III but not so in the others) as well as the timing of their assessment (during final exam period), they decided to eliminate all ambiguity by undertaking a fourth assessment that would encompass all versions at the same time (plus AESOP). The questions asked were the same six, and the test subjects were further asked to rank the systems from 1 to 4 as to how the stories produced by each version fared against the other versions vis-à-vis the same six questions asked. In this last evaluation, fAIble III clearly came out the best in all six questions.

Campfire was also extensively evaluated by human test subjects. This was done by having each test subject read two different stories. The first story (*Story #1*) was read in its entirety as generated and also told by the Campfire storyteller avatar, but was not able to be changed by the test subject. However, the test subject was able to initially specify the genre, the gender of the main character and the length of the story. The second story (*Story #2*) was the same as the first story (as originally specified by the test subject) except that the test subjects were now allowed to make changes throughout the story at their leisure. The hypotheses to be assessed through these tests were that 1) the adult test subjects will find the stories generated by Campfire to be entertaining to children; and 2) adults would find the stories to be interesting for themselves. Two surveys totalling 54 questions were given to the test subjects – one after Story #1 and the second one after Story #2. While most of the questions required a five-valued Likert scale response, 15 of these requested direct word answers. Their results indicated general success in the evaluation of the Campfire prototype.

The IPOCL planner was tested against their previous partial-order causal link (POCL) planner by assigning test subjects to read stories from one of the planners and thereafter answer questions to assess their understanding of character goals and motivations. The results showed that the IPOCL stories had improved reader comprehension about the intentions and motivations behind character goals as compared to the POCL stories.

Picture Books (PB) (Hong et al., 2009) and Picture Books 2 (PB2) (Ang et al., 2011) were also assessed by humans. The assessments for PB and PB2 were similar in a general way in that they employed a panel of three human judges. They were presented with a set of stories and were asked to say whether the stories contained a given set of criteria. A Likert scale was used to record and quantify their answers. This approach was similar to what was done for fAIble, except with much fewer test subjects assessing a greater number of stories. In the assessments of Picture Books a panel of two educators and one linguist were presented with the same 15 stories each and asked to rate each on a scale of 1–4 as to the completeness of their plots, where 4 was the highest positive rating. The story pattern evaluation rated 3.75 for the average of several other sub-criteria, while the semantic ontology was 3.33. The evaluation of grammar correctness was rated 3.29 on average of other sub-criteria). Picture Books 2 used a similar arrangement of two linguists and one story writer who were presented with 10 stories. Unfortunately, the criteria and Likert scale used were different, so a head-to-head comparison cannot be made. However, its summary of quantitative evaluation was highly positive, with values of 3.74 and 3.06 out of a possible 5 points.

The evaluation of the Suspenser system (Cheong & Young, 2015) used a generally similar experiment using test subjects as arbiters of the system's effectiveness in generating suspense at a given point in time in a story. The hypothesis was to discover any association between the story generator type and the suspense level of the story. The 98 recruited test subjects were asked to assess the level of suspense in three stories: one by Suspenser, one by a human author with high suspense, and a control narrative also done by a human author but with low suspense. Three fabulas were planned with a hierarchical partial-order causal link planner (Crossbow). Three sjuzhets were created from each of these three fabulas, one by Suspenser, and the other two by the human author – one for high suspense and the other for low suspense (the control). Each test subject was presented with three of these nine stories – one from each group, in two stages. The first part involved reading the story up to the suspense measurement point and asked to rate the suspense there on a five point Likert scale (0 = no suspense; 5 = extremely suspenseful. The second part involved showing the rest of the story, and then asked to answer some questions.

IRIS Interactive used human test subjects to play the game and these were later surveyed to determine their opinions about any suspense they felt while playing the game. The interactive version of IRIS was evaluated in the typical human test subject fashion, where 64 test subjects played a video game that contained four versions (different combinations of modules made active) of the IRIS Interactive-generated narrative. They were later asked to complete a survey asking questions about their experience and their opinions about the suspense content of the narrative.

The original Mexica system (Pérez y Pérez & Sharples, 2001) was evaluated through the typical human-based process: recruit test subjects (in their case, 50), ask them to read four stories generated by Mexica with various features, and rate them on a five-point scale on several characteristics of the story (e.g., narrative flow, coherence, suspense, etc.).

For the Mexica-Impro system, testing was done to verify that a single agent could not produce the same stories as when done by the two agents collaboratively. This was done by running the system in a guided mode where the user picked the next action to best replicate the story. The collective stories could not be replicated, showing that the collaboration of the two agents did in fact result in novel output.

Some system evaluations used two independent groups of evaluators. This is particularly relevant to interactive systems for separating the experience of working with the system from judging the quality of the created stories. The first group interacts with the system to create the stories and completes a survey to obtain their opinions; then a second group (that does not interact with the system) is subsequently used to more objectively assess the quality of the stories generated by the first group. This allows for a more objective assessment of the created stories than would be possible if, in effect their authors evaluated them. The Plan, Write and Revise system (Goldfarb-Tarrant et al., 2019) used just such a method. The system was rigorously tested with two such panels of human test subjects. The first group of 30 test subjects participated in the creation of the stories, implementing six different models of collaboration in narrative generation to varying degrees and then completed a self-reporting survey about their experience. They took part in nine experiments that ranged from no user interaction to minimal user interaction to extensive interaction. The second panel was used to only review the stories created in an arms-length type of analysis, with no regards to how they were created. The results indicated that human involvement improved the quality of the story.

STORYBOOK (Callaway & Lester, 2002) used 20 human test subjects to determine which combination of modules would generate the best narrative prose. They presented the test subjects with two pair-wise comparisons of four narratives (versions of two stories). The subjects were asked to rate them in nine *grading factors* (e.g., overall, style, grammaticality, flow, diction, readability, logicality, detail and believability) on a academic-like grading scale of A through F, where A = 4; B = 3 . . . F = 0. Unlike those for fAIble and Campfire, this evaluation did not ask qualitative questions seeking the evaluators' indication of agreement or disagreement, but rather, more specifically asked the evaluators to grade each story presented on nine different criteria.

Continuing with the discussion of Provant from section 5.1 above, the authors further assessed the system with a user study. The objective was to see if the test subjects would be able to recognise the structure of the narrative with respect to whether obstacles were introduced by the antagonist. Thirty-nine test subjects were used. The experimental design was similar to the common one – the test subjects were shown narrative plans generated by Provant and then asked questions via a self-reporting survey. The questions were to name the protagonist and the antagonist and whether the antagonist actually antagonised the protagonist by throwing up obstacles (our words) to the protagonist to keep it from achieving its goal(s). The same six narratives used in the functional tests were used here except some modifications were made to those that contained non-recoverable obstacles to remove this problem.

The character-centric neural model was evaluated using the BLEU algorithm (for evaluating machine translation quality), perplexity (for evaluating the quality of a language model), and human evaluation. Testing was done using a character-centric model and a non-character centric model to evaluate how the character embeddings affected the story. Results showed that the perplexity and character believability were better in the character centric model and human testing showed that the stories were also perceived as more reasonable.

The Dramatis system (O'Neill & Riedl, 2014) did an integrated combination of human-based and functional testing. This encompassed three separate but sequential evaluations. The first one involved the typical human-based tests – several versions of a story generated by different versions of the system are presented to human test subjects (32 in their case) and asked to rate the stories in a pair-wise fashion for their level of suspense. With the baseline human opinions about the resulting suspense curves, the two other assessments introduced changes in the Dramatis systems and created new stories with different suspense curves. These new suspense curves were then compared to those rated by the human test subjects.

Other Quantitative Metrics

Beyond the methods described above, there were reports of various quantitative metrics in the literature that could arguably be considered standard, albeit of limited scope. We describe some of these next.

Roemmele et al. **(2017)** report a comprehensive technique to evaluate linguistic quality of generated text without the need to involve human test subjects. Their objective was to assess the linguistic quality of sentences produced by two different methods – Case-based Reasoning (CBR) and Recurrent Neural Networks (RNN). Recurrent Neural Networks have been commonly used for modelling sequence data and have been applied extensively in automated language generation. They are similar to LSTMs – in fact, it can be said that LSTMs are a type of RNN. RNNs use the results of a prior iteration as partial input to the next iteration. The authors' linguistic quality assessment was tested in a sentence continuation task, where each system (CBR-based and RNN-based) was asked to predict the 21st sentence in a sequence of 20 sentences taken from a children's story. Then the sentence generated is compared to the actual 21st sentence of the actual story. The metrics used include story-independent (e.g., length of the sentences, its grammaticality, lexical diversity and lexical frequency) as well as story-dependent metrics (e.g., lexical cohesion, style matching and entity co-reference). The training data for the RNN were from existing, pre-authored stories (by human authors), but were not the same as those children's stories used for tests. The analysis was purely computational, and the evaluation did not require human judgement, which is an advantage. The results showed that the assessment method was able to distinguish between the two next-sentence creation models (CBR and RNN).

In an effort to quantify and assess how diverse stories from the same system are in relation to one another, Peinado et al. (2010) proposed an algorithm to quantitatively measure the differences between the stories. The more they differ, of course, implies that the narrative generation system produces more diverse stories. The system looks for differences between the corpora of the two stories – that is, given two stories, A and B, how different are they from each other. One of the stories (say A) is used as the benchmark, and the differences from it reflected in B are a measure of the system's novelty in creating story B. The technique proposed by Peinado et al. looks at four general elements in a story 1) the *events*, 2) the *characters;* 3) the *props;* and 4) the *scenarios* within each story. They use complex formulae that arrive at individual scores for each of these four sub-measures; a final measure that combines the four sub-measures is also computed though another formula. Their formulae involve weights that are assigned rather subjectively by the human evaluator. Although no range exists for these weights, constraints in their value are suggested, but only relative to each other (i.e., $we_1 > we_2$), where e stands for event. Hollister used this method to assess the novelty of the stories generated by Campfire. See (Hollister, 2016).

Pérez y Pérez et al. (2011) report on the EVALUATOR system – an approach (and computer system) designed to evaluate the novelty aspect of creativity of a story generated by a narrative generation system. The authors define creativity as the '*. . . creativity has to do with the generation of material that is novel with respect to the agent's knowledge base . . . and that, as a consequence, generates new knowledge-structures*' (Pérez y Pérez et al., 2011; last page of paper). So, it compares the knowledge contained in the story to the knowledge that the system had prior to creating the story. While the EVALUATOR is nominally generic (i.e., non-system specific), the system whose stories are being assessed must reflect its knowledge in specific ways for it to work. It must include a set of previous stories and a set of story actions available to it to create the stories. It considers the following aspects of a story: the sequence of events, the story-structure, and repetitive patterns. It also analyzes the structures of the new story in quantifying the novelty of the story.

In another work, Anthony Hevia, one of the student researchers in the fAIble project, wrote an unpublished internal report on the use of accepted analytical techniques to quantify the readability of the stories emerging from the various fAIble systems. Readability tests employ statistical methods and a variety of computations to quantify how readable is a corpus of text. The readability of an assessed corpus is typically given in terms of school grade level equivalent – that is, what grade level would a person have to have completed to be able to understand the text. Most of the tests reported in the literature do computations on the text corpus, such as counting words, syllables, and sentences. Readability tests are not designed to measure the semantic complexity of a corpus, nor identify its underlying structure. However, they have been extensively used in text processing applications. Hevia used an open source library known as 'textstat'[19] to obtain the Python implementations of the selected readability tests that he applied to stories generated by the three fAIble versions plus AESOP. Given that the target audience for fAIble was 6 to 8 year old children, estimating the grade level of the generated text would be important.

There are several reading tests reported in the literature, but several were discarded offhand because the stories to be assessed did not meet the pre-requisites imposed by the tests – mostly because of corpus length – or because their code was not available. Hevia selected the following three tests: 1) Flesch-Kincaid Grade Level/Flesch Reading Ease (the most popular tests) (Kincaid et al., 1975); 2) The Gunning Fox test (Gunning, 1952); 3) The Automated Readability Index (ARI) (Senter & Smith, 1967). His results showed that fAIble generated stories generally readable by its target audience.

Trends in Narrative Generation System Evaluation

In spite of the extensive evaluation reported in many of the works reviewed, no standard assessment procedure or standard tool or metrics have emerged. To be sure, several were proposed – explicitly and implicitly – but there is little evidence that these have become commonly adopted in the research community. One trend that did appear was the use of human test subjects to review a story or to participate in a video game with a generated narrative, and then rate the story (or the experience) via self-reporting surveys. While this is a natural way of evaluating the human aspect of the stories, the questions asked and how the test subjects were asked to rate the stories varied greatly from one system to the next. This is also quite expected, as the different systems had different objectives and the researchers sought to ascertain whether their system met their research objectives.

The closest thing to a standard metric was the reading tests used by Hevia – tests that already existed and were widely used for other purposes. Equally quasi-standard was the technique by Peinado et al. to computationally quantify the diversity of a story when compared to another story, presumably from the same narrative generator. Unfortunately, we did not see evidence of their wide use from our review.

Conclusion

Storytelling is a human tradition that can be traced back to humanity's origins. As advancing technology plays an exponentially greater role in our lives as it is currently expected to do, the merger of fictional narratives and technology has taken on increasing interest in the context of education, training and entertainment. This review paper surveys research related to automatically generating narratives mostly for the purpose of entertainment, although a few of the works covered are designed for education. Research in automated narrative generation systems began in the 1970s and has continued to evolve into more sophisticated systems. Our goal was to briefly discuss the historically-important works, and then followed by the recent (2010 to 2020) works in greater detail. Lastly, we included a separate section that reviews the methods used to evaluate the 'goodness' of a system, algorithm or technique.

Some of the systems reviewed represent full systems – that is, they produce ready-to-tell stories in natural language. Others address techniques that can improve an aspect of the story generation process but without producing a full story. Yet others address specific features of stories, such as the natural language used in the text of the story or the use of graphics. Some of the works reviewed are for traditional fictional stories to be read or told while others are intended for video games. Our review was categorised by three types of systems: non-interactive narrative generation systems, interactive narrative generation systems, and computer game based narrative systems (a subset of interactive narrative generation).

Non-interactive narrative systems focus on generating interesting and diverse stories that attempt to mirror human creativity. These types of systems are difficult to develop because without user input, the system must rely solely on itself to generate the stories. The encoding of the creative process and the knowledge involved is a hard task that has been addressed using different methods such as plot planning, reusing and transforming past events, stochastic selection of events, and context-centric planning. As approaches for creating story plots and generating the corresponding actions have become increasingly refined, more attention has been given to making the narrative more interesting – the foremost goal of all storytellers. Some systems have incorporated the mental states of characters, character transformations, character action failure, and the collaboration of multiple story generating agents. Lastly, the role of machine learning has increased in recent years, mostly as a way to plan the plot events but also for generating natural language text. Now that base approaches for story generation have been laid out, future work in non-interactive narrative systems can pivot their attention to tackling literary themes and devices that make human-authored stories unique and captivating. After all, as stated repeatedly in the works reviewed above, stories are much more than a sequence of actions – they must be logical (causally linked), believable and most of all, interesting.

Interactive narrative systems generate stories with the help of user interaction, either a priori or at story telling time. The level of interactivity varies widely across systems and in the ways in which users are involved in the creation process. User interaction can be as simple as picking the characters and setting of the story to be generated, or as complicated as providing real-time brain activity feedback. Developing interactive systems is in a way more difficult than non-interactive systems because the former must deal with the unpredictability of the human input, yet still compose an interesting story. There are many implementations of interactive narrative generation systems, from video games to storytelling applications and even training and educational settings. Some of the systems reviewed produce full stories while others only address aspects of the narrative generation problem. Over the last decade, systems in this category have addressed issues such as real-time adaptation to user requests, balancing between user control and the narrative presentation, and creating rich and immersive story world environments. With the advent of virtual reality, interactive storytelling can delve into new forms of user interaction. Similarly, as wearable technology becomes more popular, biofeedback as a form of user input will be more achievable.

The overriding focus of most papers reviewed was on plot planning – that is, selecting a series of character actions that are logical, believable and interesting, and that move the state of the story towards some goal. Planning systems became the natural go-to solution for this problem in the early stages of the research because they are designed to produce a sequence of events that move the state of something towards a desirable goal. However, it became evident that plots are not plans, as plots need to be logical, believable and interesting, and this means introducing intentionality, conflict, failure, suspense, character abilities and beliefs, etc. This realisation led to several variations on the traditional planning approach, as well as some non-planning approaches. As of this writing, the perfect algorithm for generating plots is still to be found.

Another area that still requires improvement, in spite of some notable recent advances, is that of natural language generation that avoids mechanical and awkward sounding sentences. In our opinion, the system that best achieved natural sounding language was the GPT-2 system. However, while it can construct short stories quite well, it is interesting that GPT-2 is not a narrative generation system to begin with, but rather, a language model. Some of the narrative generation systems we discussed have achieved quite a good level of natural language; however, this has often been at the cost of relying heavily on pre-authored story sentences that map to events. While this is the easiest solution, the work needed to create such systems results in a heavy authorial burden that can limit the story creation process. Other systems discussed generate the text for their story events on the fly, but the outcome tends to be repetitive or choppy.

A third area of needed improvement is story diversity. Many of the systems discussed take different routes to create diversity in their story plots, using one or more narrative agents writing the plot, character-driven plots, and templatized story formats. Similar to the area of natural language generation, these techniques can impose heavy authorial burden to create logical, believable, interesting <u>and</u> diverse story plots; otherwise, the stories begin to sound alike after a few stories have been generated. Nevertheless, it is clear that progress is being made towards addressing these issues.

While most of the systems reviewed focused on narrative plot generation, a few were included which did generate or had the capability to generate poetry and song. The most notable of these systems was MABLE as it was built with the sole purpose of creating ballads by transforming the narrative output of the MEXICA system using a Markov model. The GPT-2 language model is another system that can generate poetry and song through its predictive capabilities that allow it to follow the style and theme of its input. Thus, if given the start of a poem, it could continue it. Lastly, the proposed AI Stories system plans to make use of templates for poetry creation. In all cases, the automated generation of poetry and song is an area that could use increased attention.

Reduction of authorial burden was a focus of several of the systems reviewed. While some progress was made towards this goal, we are of the opinion that in the short to medium term, systems will require voluminous knowledge about the world in order to produce stories with the same depth and breadth that human authors do. Data-based approaches such as with machine learning may well solve this problem. However, nothing that we saw suggests that this is something imminent.

As a word of caution, we should note that automated narrative generation systems of the type reviewed here could conceivably be misused to compose 'fake news' and passed as true stories. Such fake news have been made notable in recent years, as false stories have been published in unscrupulous Internet blogs by both ends of the political spectrum in an effort to mislead the readers and sow dissent. Work is being done at creating pipelines to detect such fake narratives, such as the work by Tangherlini et al. (2020). At some point in time, we hope that sufficient progress is made in algorithms that detect and identifies such false narratives for what they are.

In closing, the advances made in automated story generation since 2010 have made clear progress towards the goal of becoming more dynamic and diverse. While there is still much work to be done in this area, these systems are the stepping stones towards the coming future of true and complete automated story generation.

Notes

1. Fabula comes from Russian romantic narratology to mean the sequence of events in the story. It contains all the events in a story in the proper sequence.
2. The term sjuzhet also comes from Russian romantic narratology to mean what parts of the fabula are revealed to the reader as part of the telling of the story.
3. Classical AI planning research began in the late 1960s with the STRIPS system. It sought to find a sequence of actions that would transform the world from an initial state to a goal state. It combined theorem proving through resolution with heuristic state-space searches. It assumes complete knowledge about and control over the world, and that all actions are deterministic in nature. See (Fikes and Nilsson, 1993) for a brief introduction to AI planning systems.
4. HTNs compose a plan as a set of primitive tasks whose sequence is governed by a network of constraints. It has some rough similarity to STRIPS.
5. Partial-order planning is a planning strategy based on least-commitment planning, where only essential ordering is decided and all other ordering is treated flexibly (Weld, 1994). "The idea of a partial-order planner is to have a partial ordering between actions and only commit to an ordering between actions when forced." (Poole and Mackworth, 2017).
6. LSTMs are a form of neural networks that use a constant error flow to bridge over gaps in time-based learning, such as would exist in composing narratives. See (Hochreiter and Schmidthube, 1997) for further information.
7. Haslum defines compilation as "*. . . a systematic remodeling of the problem such that a classical plan for the reformulated problem meets also the non-classical requirements.*" (Haslum, 2012, p. 383),
8. Briefly, CBR looks for similarities between a current problem and historical problems in memory (in the form of *cases*) that were once successfully resolved. A solution that resolved a similar problem in the past can be applied to the current problem if the problems are sufficiently similar. See (Kolodner, 1993) for a full description of CBR.
9. The ramification problem is when an action taken has unintended secondary or tertiary consequences or effects that were not envisioned.
10. Constructivist Theory – learning theory in education which states that the learner (or reader in this case) understands material based on their own unique experiences (Western Governors University, 2020).
11. For further information on Cased-Based Planning, refer to Hammond (1986) and Kolodner (1993).
12. Plans are defined as a set of primitive actions in the world. Complete knowledge of the world is necessary.
13. State-space Planning – process for searching for a solution from a state space, which contains all of the data to be searched; the resulting plan is a path through the state space (Nau, 2012)

14. BDI is a common paradigm for creating intelligent software agents. See Bratman (1992). It assumes that agents have three mental attitudes: beliefs (knowledge about the world); desires (goals, and plans to achieve these goals); and intentions (commitment to pursue the desires).
15. CxBR is a context-centric paradigm for intelligent reasoning by agents in tactical simulations. See (Gonzalez, 2014) for a full description of Context-Based Reasoning.
16. That is, they work as a result of a complex set of computations with their weights, but it is not normally clear how the weights affect their solutions.
17. Q-Learning is a reinforcement learning algorithm that is values-based and model-free and works by learning the value of the optimal policy independent of the agent actions (Shyalika, 2019). Reinforcement learning, in general uses trial and error to find a behavior (called a policy) by rewarding moves by the learning agent that lead to success and punishing those that result in failure.
18. Sequence-to-sequence neural networks are deep networks that can learn sequences of words or images of arbitrary lengths, something that mainstream deep neural networks cannot easily do. See (Sutskever, Vinyals & Le, 2014)
19. https://github.com/shivam5992/textstat

Acknowledgments

The compiling and writing of this paper was partially supported by the US National Science Foundation under their International Research Experience for Students (IRES) program via grant #1458272.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the Office of International Science and Engineering [ISE-1458272].

References

Alvarez, M. J., Amaya, R. E., Benko, K. A., Martin, J. T., Knauf, R., Jantke, K. P., & Gonzalez, A. J., (2019) Hello, Narratives: Character Development in Automated Narrative Generation. *Proceedings of the 32nd Annual Florida Artificial Intelligence Research Society Conference (FLAIRS-2019), Sarasota, FL*. AAAI Press.

Ammanabrolu, P., Cheung, W., Broniec, W., & Riedl, M. O. (2020) *Automated Storytelling via Causal, Commonsense Plot Ordering*. arXiv preprint arXiv:2009.00829.

Ang, K., Yu, S., & Ong, E. (2011) Theme-Based Cause-Effect Planning for Multiple-Scene Story Generation. *Proceedings of the Second International Conference on Computational Creativity*, Mexico City. https://computationalcreativity.net/iccc2011/proceedings/the_narrative/ang_iccc11.pdf

Aylett, R. S., Louchart, S., Dias, J., Paiva, A., & Vala, M. (2005, September). FearNot!–an experiment in emergent narrative. In Panayiotopoulos T., Gratch J., Aylett R., Ballin D., Olivier P., Rist T. (Eds.), *International workshop on intelligent virtual agents* (pp. 305–316). Berlin, Heidelberg: Springer. https://doi.org/10.1007/11550617_26

Bailey, P. (1999) Searching for storiness: Story-generation from a reader's perspective. In *Working notes of the Narrative Intelligence Symposium*. (pp. 157–164), North Falmouth, MA: AAAI Press.

Bates, J. (1992). *The nature of characters in interactive worlds and the Oz project*. School of Computer Science, Carnegie Mellon University.

Bayon, V., Wilson, J. R., Stanton, D., & Boltman, A. (2003). Mixed reality storytelling environments. *Virtual Reality 7* (1), 54–63. https://doi.org/10.1007/s10055-003-0109-6

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). *COMET: Commonsense transformers for automatic knowledge graph construction*. arXiv preprint arXiv:1906.05317.

Bottoni, B., Moolenaar, Y., Hevia, A., Anchor, T., Benko, K., Knauf, R., Jantke, K. P., Gonzalez, A. J., & Wu, A. S. (2020). Character Depth and Sentence Diversification in Automated Narrative Generation. *Proceedings of the 33rd[d] Annual Florida Artificial Intelligence Research Society Conference (FLAIRS-2020),* North Miami Beach, FL: AAAI Press.

Boulenger, V., Roy, A. C., Paulignan, Y., Deprez, V., Jeannerod, M., & Nazir, T. A. (2008). Crosstalk between language processes and overt motor behavior in the first 200 msec of processing. *Journal of Cognitive Neuroscience*, *18*(10), 1607–1615. https://doi.org/10.1162/jocn.2006.18.10.1607

Bower, G. H., & Clark, M. C. (1969). Narrative stories as mediators for serial learning. *Psychonomic Science*, *14*(n 4), 181–182. https://doi.org/10.3758/BF03332778

Bratman, M. E. (1992). Shared Cooperative Activity. *The Philosophical Review*, *101*(n 2), 327–341. https://doi.org/10.2307/2185537

Breault, V., Ouellet, S., & Davies, J. (2021). Let CONAN tell you a story: Procedural quest generation. *Entertainment Computing, 38.*, http://arxiv.org/abs/1808.06217

Brenner, M. (2010) Creating dynamic story plots with continual multiagent planning. *Proceedings of the 24th AAAI National Conference on Artificial Intelligence, Atlanta, GA*. AAAI Press

Brézillon, P. (2004). Representation of Procedures and Practices in Contextual Graphs. *The Knowledge Engineering Review*, *18*(n 2), 147–174. https://doi.org/10.1017/S0269888903000675

Burtenshaw, B. (2020) *AI Stories: An Interactive Narrative System for Children*. arXiv preprint arXiv:2011.04242.

Caine, R. N., & Caine, G. (1991). *Making Connections: Teaching and the Human Brain*. Association for Supervision and Curriculum Development.

Callaway, C. B., & Lester, J. C. (2002). Narrative prose generation. *Artificial Intelligence*, *139*(n 2), 213–252. https://doi.org/10.1016/S0004-3702(02)00230-8

Chang, H., & Soo, V. (2008). Planning to Influence Other Characters in Agent-Based Narratives. In *Integrating Technologies for Interactive Stories Workshop, International Conference on Intelligent Technologies for Interactive Entertainment*(pp. 12-17).

Cheong, Y. G., & Young, R. M. (2015). Suspenser: A story generation system for suspense. *IEEE Transactions on Computational Intelligence and AI in Games*, *7*(1), 39–52. https://doi.org/10.1109/TCIAIG.2014.2323894

Ciarlini, A. E. M., Pozzer, C. T., Furtado, A. L., & Feijo, B. (2005) A logic-based tool for interactive generation and dramatization of stories. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*(pp. 133-140), Valencia, Spain. Association for Computing Machinery

Colby, B. N. (1973). A Partial Grammar of Eskimo Folktales. *American Anthropologist*, *75*(n 3), 645–662. https://doi.org/10.1525/aa.1973.75.3.02a00010

Coman, A., & Munoz-Avila, H. (2012) Creating Diverse Storylines by Reusing Plan-Based Stories. *Working Notes of the ICCBR-12 Workshop on TRUE and Story Cases: Traces for Reusing Users' Experiences - Cases, Episodes, and Stories, Lyon, France*. Springer.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence*, *41*(1), 1–63. https://doi.org/10.1016/0004-3702(89)90077-5

Fendt, M. W., & Young, M. (2011) The case for intention revision in stories and its incorporation into IRIS — A story-based planning system. *Proceedings of the 18th AIIDE Conference on Intelligent Narrative Technologies* (pp. 10–16), Stanford, CA. AAAI Press.

Fendt, M. W., & Young, R. M. (2014) Adapting IRIS, a Non-Interactive Narrative Generation System, to an Interactive Text Adventure Game. *Proceedings of the Florida Artificial Intelligence Society Conference (Flairs* 2014), Pensacola Beach, FL: AAAI Press.

Fikes, R. E., & Nilsson, N. J. (1993). STRIPS A Retrospective. *Artificial Intelligence*, *59*(1–2), 227–232. https://doi.org/10.1016/0004-3702(93)90190-M

Gilroy, S. W., Porteous, J., Charles, F., Cavazza, M., Soreq, E., Raz, G., Ikar, L., Or-Borichov, A., Ben-Arie, U., Klovatch, I., & Hendler, T. 2013 A Brain-Computer Interface to a Plan-Based Narrative. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI-2013)*, pp. 1997–2005, Beijing, China: AAAI Press.

Goldfarb-Tarrant, S., Feng, H., & Peng, N. (2019) *Plan, Write, and Revise: An Interactive System for Open-Domain Story Generation*. arXiv preprint arXiv:1904.02357.

Gonzalez, A. J. (2014). Tactical Reasoning through Context-Based Reasoning. In P. Brezillon & A. J. Gonzalez (Eds.), *Context in Computing: A Cross-disciplinary Approach for Modeling the Real World (pp. 491-508)*. Springer Science + Business Media.

Grinblat, J., & Bucklew, C. B. (2017) Subverting historical cause & effect: Generation of mythic biographies in Caves of Qud. *Proceedings of the 12th International Conference on the Foundations of Digital Games* (pp. 1–7), Hyannis, MA: Association for Computing Machinery.

Gunning, R. (1952). *The Technique of Clear Writing* (36–37). McGraw-Hill.

Hammond, K. J. (1986) CHEF: A model of case-based planning. *Proceedings of the American Association for Artificial Intelligence (AAAI) National Conference* (pp. 267–271), Philadelphia, PA: AAAI Press.

Haslum, P. (2012). Narrative Planning: Compilations to Classical Planning. *Journal of Artificial Intelligence Research*, *44*(1), 383–395. https://doi.org/10.1613/jair.3602

Hochreiter, S., & Schmidhuber, J. (1997). Long Short Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hollister, J. R. (2016) A Contextual Approach to Real Time, Interactive Narrative Generation. [*PhD Dissertation*. University of Central Florida, Orlando, FL]. STARS. http://purl.fcla.edu/fcla/etd/CFE0006687

Hollister, J. R., & Gonzalez, A. J. (2018). The Cooperating Context Method. *Modelisation et Utilisation du Contexte. iSTE OpenScience,*(1),18-2. https://doi.org/10.21494/ISTE.OP.2018.0232

Hollister, J. R., & Gonzalez, A. J. (2019). The CAMPFIRE Storytelling System – Automatic Creation and Modification of a Narrative. *Journal of Experimental and Theoretical Artificial Intelligence*, *31*(1), 15–40. https://doi.org/10.1080/0952813X.2018.1517829

Hong, A. J., Solis, C., Siy, J. T., Tabirao, E., & Ong, E. (2009) Planning Author and Character Goals for Story Generation. *Proceedings of the NAACL Human Language Technology Workshop on Computational Approaches to Linguistic Creativity* (pp. 63–70), Boulder CO, June 2009. Association for Computational Linguistics.

Kahn, K. M. (1979) Creation of computer animation from story descriptions. [*Ph.D. Dissertation*, Massachusetts Institute of Technology, Cambridge, MA]. MIT Libraries. http://hdl.handle.net/1721.1/6875

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 Florida Artificial Intelligence Research Society Conference (Flairs-31)*, (pp. 282–287), Melbourne, FL: AAAI Press.

Kelly, J. P., Botea, A., & Koenig, S. (2008) Offline Planning with Hierarchical Task Networks in Video Games. *Proceedings of the AIIDE Conference* (pp. 60–65), Stanford, CA: AAAI Press.

Kincaid, J. P., Fishburne, R. P., Rogers, R. L., & Chissom, B. S. (1975) *Derivation of new readability formulas (automated readability index, fog count, and Flesch reading ease formula) for Navy enlisted personnel*. Research Branch Report 8–75. Chief of Naval Technical Training: Naval Air Station Memphis

Kolodner, J. L. (1993). An introduction to case-based reasoning. *Artificial Intelligence Review*, *6*(1), 3–34. https://doi.org/10.1007/BF00155578

Lebowitz, M. (1985). Story-telling as planning and learning. *Poetics*, *14*(n 6), 483–502. https://doi.org/10.1016/0304-422X(85)90015-4

Liu, D., Li, J., Yu, M. H., Huang, Z., Liu, G., Zhao, D., & Yan, R. (2020) A character-centric neural model for automated story generation. *Proceedings of the AAAI Conference on Artificial Intelligence* vol 34, pp. 1725–1732, New York, NY: AAAI Press.

Liu, H., & Singh, P. (2002). MAKEBELIEVE: Using commonsense knowledge to generate stories. In *AAAI/IAAI. pp* (pp. 957–958). AAAI Press. https://doi.org/10.5555/777092.777241

Liu, H., & Singh, P. (2004). ConceptNet — A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal*, *22*(n 4), 211–226. https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d

Machado, I., & Paiva, A. (1999) Heroes, Villains, Magicians, . . . : Believable Characters in a Story Creation Environment. *AI-ED '99 Workshop on Animated and Personified Pedagogical Agents*. pp. 39, Le Mans, France: CiteSeer.

Mateas, M. (2002) Interactive Drama, Art and Artificial Intelligence. [*Ph.D. Dissertation*. Carnegie-Mellon University, Pittsburgh, PA]. Carnegie Mellon University

Mateas, M., & Stern, A. (2005) Structuring content in the *Façade* interactive drama architecture. *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference*. pp. 93–98, Marina del Rey, CA: AAAI Press.

Mateas, M., Vanouse, P., & Domike, S. (2000, July). Generation of ideologically-biased historical documentaries. In *AAAI/IAAI* (pp. 236–242). AAAI Press.

McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., & Wilkins, D. (1998) *PDDL - the Planning Domain Definition Language*. Technical report, CVC TR-98-003/DCS TR-1165, Yale University, 1998. CiteSeerX

Meehan, J. (1977) TALE-SPIN: An Interactive Program that Writes Stories. *Proceedings of the 5th International Joint Conference on Artificial Intelligence*. pp. 91–98, Cambridge, MA: Morgan Kaufmann Publishers Inc.

Meo, T., Raghavan, A., Salter, D. A., Tozzo, A., Tamrakar, A., & Amer, M. R. (2018) Aesop: A Visual Storytelling Platform for Conversational AI. *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18)*. pp. 5844–5846, Stockholm, Sweden: AAAI Press.

Nau, D. (2012, February) *Chapter 4 State-Space Planning*. Lecture Slides. University of Maryland. https://www.cs.umd.edu/~nau/planning/slides/chapter04.pdf

Niehaus, J., & Young, R. M. (2010) A method for generating narrative discourse to prompt inferences. *Proceedings of the Intelligent Narrative Technologies III Workshop*, Monterey, California. Association for Computing Machinery.

O'Neill, B., & Riedl, M. (2014) Dramatis: A computational model of suspense. *Proceedings of the 28th AAAI National Conference on Artificial Intelligence, Quebec City, Canada*: AAAI Press.

Ontañón, S., & Zhu, J. (2010) Story and text generation through computational analogy in the Riu system. *Proceedings of the Sixth Artificial Intelligence and Interactive Digital Entertainment Conference, Stanford, CA*: AAAI Press.

Paul, A. M. (2012) Your Brain on Fiction. New York Times. A. G. Sulzberger

Peinado, F., Francisco, V., Hervás, R., & Gervás, P. (2010). Assessing the Novelty of Computer-Generated Narratives Using Empirical Metrics. *Minds and Machines*, *20*(4), 565–588. https://doi.org/10.1007/s11023-010-9209-8

Perez y Pérez, P. (2015). A computer-based model for collaborative narrative generation. *Cognitive Systems Research*, *36*(C), 30–48. https://doi.org/10.1016/j.cogsys.2015.06.002

Pérez y Pérez, R., Ortiz, O., Luna, W., Negrete, S., Castellanos, V., Peñalosa, E., & Ávila, R. (2011) A System for Evaluating Novelty in Computer Generated Narratives. *Proceedings of the Second International Conference on Computational Creativity*, Mexico City. computationalcreativity.net

Pérez y Pérez, R., & Sharples, M. (2001). MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, *13*(2), 119–139. https://doi.org/10.1080/09528130010029820

Phillips, M. A. (n.d.) *Dramatica*: The Next Chapter in Story Development. Write Brothers, Inc. https://dramatica.com/.

Pickett, G., Khosmood, F., & Fowler, A. (2015) Automated generation of conversational non player characters. *Proceedings of the 11th Artificial Intelligence and Interactive Digital Entertainment Conference, Santa Cruz, CA*: AAAI Press.

Poole, D., & Mackworth, A. (2017). *Artificial Intelligence: Foundations of Computational Agents* (second edition ed.). Cambridge University Press.

Porteous, J., Cavazza, M., & Charles, F. (2010). Applying planning to interactive storytelling. *ACM Transactions on Intelligent Systems and Technology*, *1*(2), 1–21. https://doi.org/10.1145/1869397.1869399

Porteous, J., Charles, F., & Cavazza, M. 2013 NetworkING: Using character relationships for interactive narrative generation. *Proceedings of the AAMAS 2013 Conference*. pp. 595–602, St. Paul, MN: International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC

Porteous, J., & Lindsay, A. (2019). Protagonist vs. *Antagonist PROVANT: Narrative Generation as Counter Planning. AAMAS, (2019)*, 1069–1077, Montreal, Canada: International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC. https://dl.acm.org/doi/10.5555/3306127.3331805

Radford, A., Wu, J., Amodei, D., Clark, J., Brundage, M., & Sutskever, I. (2019b) *Better Language Models and Their Implications*. OpenAI. Retrieved from January 18, 2021 https://openai.com/blog/better-language-models/

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019a). *Language Models are Unsupervised Multitask Learners*. OpenAI blog.

Riedl, M., Saretto, C. J., & Young, R. M. (2003) Managing interaction between users and agents in a multi-agent storytelling environment. *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*. Melbourne, Australia, July 14-18

Riedl, M. O. (2010). Story Planning: Creativity through Exploration, Retrieval, and Analogical Transformation. *Minds & Machines*, *20*(4), 589–614. https://doi.org/10.1007/s11023-010-9210-2

Riedl, M. O., & Young, R. M. (2010). Narrative Planning: Balancing Plot and Character. *Journal of Artificial Intelligence Research*, *39 (1)*, 217–268. https://doi.org/10.1613/jair.2989

Roemmele, M., Gordon, A. S., & Swanson, R. (2017) Evaluating story generation systems using automated linguistic analyses. *Proceedings of the SIGKDD 2017 Workshop on Machine Learning for Creativity*. *pp*. 13–17, Halifax, Canada: Association for Computing Machinery.

Rouse, M., & Lutkevich, B. (2020) *Language modeling*. TechTarget. Retrieved from January 18, 2021, https://searchenterpriseai.techtarget.com/definition/language-modeling

Rowe, J., Mott, B., McQuiggan, S., Robison, J., Lee, S., & Lester, J. (2009) Crystal island: A narrative-centered learning environment for eighth grade microbiology. *Workshop on intelligent educational games at the 14th international conference on artificial intelligence in education*. Brighton, UK. pp. 11–20. IOS Press.

Rowe, J. P., Shores, L. R., Mott, B. W., & Lester, J. C. (2010) A framework for narrative adaptation in interactive story-based learning environments. *Proceedings of the Intelligent Narrative Technologies III Workshop, Monterey, CA*: Association for Computing Machinery

Rumelhart, D. E. (1975). Notes on a schema for stories. *Paper presented at the Representation and Understanding Conference*, Orlando, FL: Elsevier Inc. https://doi.org/10.1016/B978-0-12-108550-6.50013-6

Senter, R. J., & Smith, E. A. (1967) *Automated Readability Index*. Technical Report, Wright-Patterson Air Force Base: iii. AMRL-TR-6620.

Short, E. (2016) *Nothing for Dinner (Szilas, N. IDtension) [Web log review]*. Emily Short's Interactive Storytelling. Retrieved from January 18, 2021, https://emshort.blog/2016/09/05/nothing-for-dinner-nicolas-szilas-et-al-idtension/

Shyalika, C. (2019) *A Beginners Guide to Q-Learning*. Towards Data Science Inc. Retrieved from January 27, 2021, https://towardsdatascience.com/a-beginners-guide-to-q-learning-c3e2a30a653c

Singh, D., Ackerman, M., & Perez Y Perez, R. (2017) *A ballad of the Mexicas: Automated lyrical narrative writing*. ICCC.

Singh, P. (2002) The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. Palo Alto, CA: AAAI.

Spierling, U. (2005) Interactive Digital Storytelling: Towards a Hybrid Conceptual Approach. *DiGRA International Conference 2005: Changing Views: Worlds in Play, Vancouver, Canada*: DiGRA.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014) *Sequence to sequence learning with neural networks*. arXiv preprint arXiv:1409.3215.

Szilas, N. (2003) IDtension: A narrative engine for Interactive Drama. *Proceedings of the technologies for interactive digital storytelling and entertainment (TIDSE) conference*, vol 3, pp. 1–11, Darmstadt, Germany: Springer-Verlag, Berlin, Heidelberg.

Szilas, N., Richle, U., Dumas, J., Boggini, T., & Habonneau, N. (2015) *Nothing for Dinner*. Universite de Geneve. http://nothingfordinner.org/portal/

Tangherlini, T. R., Shahsavari, S., Shahbazi, B., Ebrahimzadeh, E., Roychowdhury, V., & Lin, Y.-R. (2020). An automated pipeline for the discovery of conspiracy and conspiracy theory narrative frameworks: Bridgegate, Pizzagate and storytelling on the web. *PloS One*, *15*(6), e0233879. https://doi.org/10.1371/journal.pone.0233879

Tearse, B., Mateas, M., & Wardrip-Fruin, N. (2010) MINSTREL Remixed: A rational reconstruction. *Proceedings of the Intelligent Narrative Technologies III Workshop, Monterey , CA*: Association for Computing Machinery.

Thalhofer, F. (n.d.) *KORSAKOW* 6. Korsakow. http://www.korsakow.com/.

Thorne, B. R., & Young, R. M. (2017) Generating Stories that Include Failed Actions by Modeling False Character Beliefs. *Proceedings of the AIIDE-17 Workshop on Intelligent Narrative Technologies*. pp. 244–251, Snowbird, UT: AAAI Press.

Turner, S. R. (1991) A case-based model of creativity. *Annual Conference of the Cognitive Science Society*, vol 13, pp. 933–937, Hillsdale, NJ: Psychology Press

Vannini, N., Enz, S., Sapouna, M., Wolke, D., Watson, S., Woods, S., Dautenhahn, K., Hall, L., Paiva, A., Andre, E., Aylett, R., & Schneider, W. (2011). "FearNot!": A computer-based anti-bullying programme designed to foster peer intervention. *European Journal of Psychology of Education*, *26*(n 1), 21–44. https://doi.org/10.1007/s10212-010-0035-4

Veale, T. (2014) Coming Good and Breaking Bad: Generating Transformative Character Arcs for Use in Compelling Stories. *5th International Conference on Computational Creativity (ICCC)*, Ljubljana, Slovenia. computationalcreativity.net

Venour, C., & Reiter, E. (2008) *A Tutorial for SimpleNLG*. University of Aberdeen. http://www.csd.abdn.ac.uk/~ereiter/simplenlg

Wade, J., Wong, J., Waldor, M., Pasqualin, L., Jantke, K. P., Knauf, R., & Gonzalez, A. J. (2017) A stochastic approach to character growth in automated narrative generation. *Proceedings of the 2017 Florida Artificial Intelligence Research Society Conference (Flairs 2017), Marco Island, FL: AAAI Press*.

Wang, P., Rowe, J., Min, W., Mott, B., & Lester, J. (2017) Interactive Narrative Personalization with Deep Reinforcement Learning. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. pp. 3852–3858, Melbourne, Australia: AAAI Press.

Ware, S. G., & Young, R. M. (2011) CPOCL: A narrative planner supporting conflict. *Proceedings of the AIIDE, Stanford, CA: AAAI Press.*

Weld, D. S. (1994). An Introduction to Least Commitment Planning. *AI Magazine*, *15*(4), 27. https://doi.org/10.1609/aimag.v15i4.1109

Western Governors University. (2020) *What Is Constructivism? Western Governors University*. https://www.wgu.edu/blog/what-constructivism2005.html.

Weyhrauch, P. W. (1997) Guiding Interactive Drama. *Ph.D. Dissertation*. Carnegie Mellon University, Pittsburgh, PA.

Willingham, B. (2013). *The Wolf Among Us [Computer software]*. Telltale Games.

Yao, L., Peng, N., Ralph, W., Knight, K., Zhao, D., & Yan, R. (2019) Plan-and-write: Towards better automatic storytelling. *Proceedings of the thirty-third AAAI National Conference on Artificial Intelligence (AAAI), Honolulu, HI*: AAAI Press.

Young, R. M. (2000) Creating Interactive Narrative Structures: The Potential for AI Approaches. *Proceedings of the AAAI Spring Symposium on Artificial Intelligence and Computer Games*. AAAI Press.