

Story and Text Generation through Computational Analogy in the Riu System

Santiago Ontañón

Artificial Intelligence Research Institute (IIIA-CSIC)
Spanish Council for Scientific Research (CSIC)
Campus UAB, 08193 Bellaterra, Spain
santi@iiia.csic.es

Jichen Zhu

Department of Digital Media,
University of Central Florida,
Orlando, FL, USA 32826-3241
jzh@mail.ucf.edu

Abstract

A key challenge in computational narrative is story generation. In this paper we focus on analogy-based story generation, and, specifically, on how to generate both story and text using analogy. We present a dual representation formalism where a human-understandable representation (composed of English sentences) and a computer-understandable representation (consisting in a graph) are linked together in order to generate both story and natural language text by analogy. We have implemented our technique in the Riu interactive narrative system.

Introduction

As a key challenge in computational narrative, story generation needs to address both technical and aesthetic challenges. Automatic story generation has spurred interest in, at least, two fields: artificial intelligence and electronic literature. On the one hand, the complex domain of story generation requires developments from various AI fields. On the other hand, computational narrative is identified as a potential art form of our time. In its full potential, computer generated stories should be able to depict a wide spectrum of human conditions and expressions with the same breadth and depth as traditional narratives (Murray 1998).

The AI community's emphasis on content generation contrasts with the electronic literature (e-lit) community's focus on the aesthetics of the final text output. Most AI approaches to story generation focus on generating some symbolic description of a story, and text generation is just achieved through a post-process. On the other hand, electronic literature productions, like *The Policeman's Beard is Half-constructed* may be controversial in terms of whether the computer program (e.g. *Ractor*) actually generated the content, but the text is no doubt intriguing and pleasantly surprising to a reader. With a handful of exceptions, the AI computational narrative / story generation community does not communicate as much as they could because of different value systems and methodology. We believe that the concern for the final text is a useful way to guide our algorithmic exploration. To this end, in this paper, we present an analogy-based story generation technique, in which, text generation

is not a post-process of story generation, but is achieved in parallel with the process of content generation, giving equal emphasis to content and text generation.

Specifically, we will present an extension to the Riu computational narrative system which is able to generate stories by extending a short story by analogy with another, longer, story. Computational analogy methods are very sensitive to the way data is represented, and therefore, we make special emphasis on how can stories be represented for successful analogy-based story generation. The main contributions of this paper are: a) an approach for story generation through analogy for interactive narrative, b) a method for generating stories that does not decouple "content generation" from "text generation" by having a dual representation combining human-understandable with computer-understandable parts.

The work presented in this paper builds on top of our previous work (Zhu and Ontañón 2010a) on analogy-based story generation. Where the contribution of this paper is that in previous work we had exclusively focused on content generation, disregarding text generation.

The remainder of this paper is organized as the following. First, we introduce the Riu system. Then we will focus on story representation, introducing our key contribution in story representation: a dual representation linking a computer-understandable representation and a human-understandable representation. After that, we will present our approach for story generation. Finally we will present some experimental results, and conclude with related work and conclusions.

Riu

Riu is a text-based interactive narrative system that uses analogy-based story generation to narrate stories about a robot character Ales, who has initially lost his memories. Throughout the story, Ales constantly oscillates between his recovering memory world and the main story world. Events happening in the memory world may impact the development in the real world. Riu explores the same story world as Memory, Reverie Machine (MRM) (Zhu 2009). While MRM was developed on the GRIOT system's conceptual blending framework (Harrell 2007), Riu focuses on computational analogy with a force-dynamics-based story representation. And specifically, it uses the *Structure Mapping Engine* (SME) (Falkenhainer, Forbus, and Gentner 1989) to

One day, Ales was walking in an alley.
 when he saw a cat in front of him.
Ales used to play with a bird when he was young.
Ales was very fond of it.
But the bird died, leaving ALES really sad.
 Ales hesitated for a second about what to do with the cat.
 (IGNORE PLAY FEED)
 > **play**
 Ales did not want to PLAY because he did not want
 CAT to be DEAD or ALES to be SAD
 (FEED IGNORE)
 > **ignore**
 Ales just kept walking.
 The cat walked away.
 ...

Figure 1: An interaction with Riu. Italics represents Ales’s past memories, and bold text represents user input.

achieve a higher level of generativity than MRM.

Informed by stream of consciousness literature such as *Mrs. Dalloway* (Woolf 2002 1925), Riu’s goal is to depict characters’ inner memory world through its correlation with the real, physical world. In this paper we focus on the use of analogy as well as text generation in Riu.

Figure 1 shows a sample interaction with Riu. The story starts with Ales’ encounter of a cat in the street while going to work, which triggers his memory of a previous pet bird in a “flashback”. There are three possible actions Ales can take at this point — “ignore,” “play” with, or “feed” the cat. In this example, the user first chooses “play.” However, the strong similarity between “playing with the cat” and “playing with his pet bird” leads to an analogical mapping and the subsequent (naive) inference that “if Ales plays with the cat, the cat will die and he will be very sad.” Hence Ales refuses to play with the cat and the system removes this action. The story continues after the user selects “ignore.” Notice, that the text generated by Riu after the user selects the action “play,” which corresponds to the part where Riu has used analogy to generate what could happen if Ales plays with the cat, looks very mechanical. The technique reported in this paper corresponds to an alternative, experimental, analogy-based story generation technique, which can generate text at the same time as the story is generated.

The Riu system contains a pre-authored main story, which is represented as a graph, where each node is a scene and each link is a possible action that Ales may take. Additionally, Riu contains a repository of lost memories of Ales, each of which is represented also as a scene. Scenes are the basic story unit used in Riu (see next section). When the main character Ales faces a decision point (i.e. when it has to choose one among several actions to execute), Ales imagines what would happen if each action is executed. This imagination process uses computational analogy in the following way: given the current scene s and an action a , Ales finds the most similar past memory m , and predicts what would happen in s after a is executed by analogy with what happened in memory m . The remainder of this paper explains how this analogy-based story generation process works.

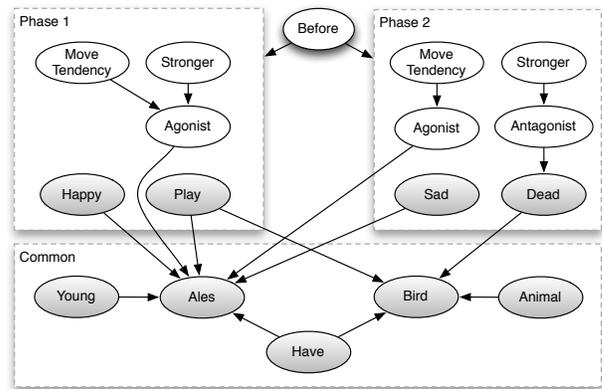


Figure 2: The Computer-understandable description of a scene in Riu

Story Representation in Riu

Stories in Riu are composed of collections of *scenes*. For us, a scene is a small encapsulated piece of a story, which happens in a single location. A scene is the primitive story representation that Riu manipulates. Given that both the user and Riu have to understand scenes, a scene contains two basic pieces: (HuD), a *Computer-understandable description* (CuD), and a *Human-understandable description*

Computer-Understandable Description

The computer-understandable description, or CuD, consists of a graph G divided in *phases*. Each node in the graph represents an object (e.g. a book), character (e.g. Ales), action (e.g. play), relation (e.g. friends), or attribute (e.g. young) in the scene. Links in the graph link actions, relations and properties with objects and characters. For instance, if we have a “play” node, it can be linked to two characters, meaning that they play together. The set of nodes in the graph is divided into a sequence of phases. Each phase corresponds to a different point in the temporal line of the scene. Each scene has a special group called *common*, containing the nodes that are common to all the phases.

For the purposes of story generation, it is important for Riu to be able to find deep analogies among source and target scenes. For that purpose, in Riu, we annotate each scene using *force dynamics*. Leonard Talmy (1988) defined *force dynamics* as way to represent the semantics of sentences using the concept of “force.” Thus, in addition to all the nodes representing objects, actions, etc. Riu includes a set of nodes corresponding to *force dynamics* annotations (see below), which are helpful to identify deep analogies between scenes.

A basic force dynamics pattern contains two entities, an *Agonist* (the focal entity) and an *Antagonist*, exerting force on each other. An entity has a *tendency* towards either motion/action or rest/inaction, and the stronger entity manifests its tendency at the expense of its opposer. Force dynamics describes not only physical forces, but also *psychological and social interactions*, conceiving such interactions as psychological “pressure.” Force dynamics allows Riu to find deeper analogies among the scenes, since it gives us a sys-

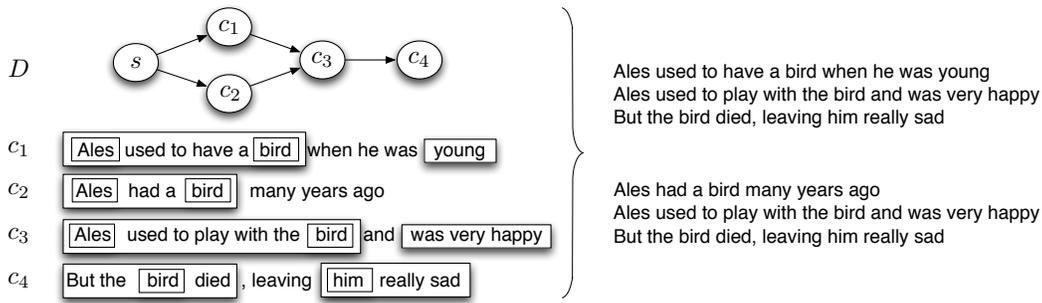


Figure 3: The Human-understandable description of a scene in Riu.

tematic way to annotate each scene allowing the computational analogy method of Riu to identify the agonist in one scene with the agonist in another scene, regardless of whether these two agonists share any surface similarities.

Figure 2 shows an example of the CuD of a scene, corresponding to the same scene depicted in Figure 3. We can see that the scene is divided in two phases: in the first phase, Ales is happy and plays with a bird, and in the second phase the bird is dead and Ales is sad. White nodes in the figure correspond to the force dynamics annotations.

Human-Understandable Description

The human-understandable description, or HuD, is composed of a series of English sentences $C = \{c_1, \dots, c_n\}$. And a finite state machine D , which determines the order in which the sentences can be combined to form the text description of a scene. D is useful, since we would not like Riu to produce the exact same text to describe a given scene each time the system is run. Thus, Riu contains alternative sentences for the different happenings of each scene, which are combined into a coherent description using D .

Figure 3 shows an example of the HuD of a scene including the finite state machine and the set of English sentences. In the example, D specifies that, starting from an initial state s , the possible orders in which the sentences can be combined are: c_1, c_3, c_4 and c_2, c_3, c_4 .

Linking the Hud to the CuD

Finally, the HuD and the CuD of a scene are not independent. Nodes in the CuD graph are linked to pieces of the HuD sentences. Figure 4 shows an illustration of how one of the sentences from Figure 3 is related to some of the nodes in the graph from Figure 2.

Thanks to these links between the CuD and the HuD, when Riu manipulates a scene graph, it also knows how to manipulate the English sentences accordingly.

Thus, a scene s in Riu is represented as a triplet: HuD (English sentences plus a finite state machine to compose them), CuD (represented by a graph), and the list of links between nodes in the graph and parts of the sentences.

Analogy-based Story Generation

In Riu, the computational analogy-based story generation process is used to implement Ales' imagination. Given the

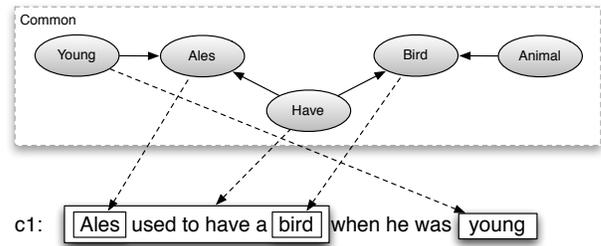


Figure 4: The English sentences are linked to nodes in the graph which represents every scene.

current scene T that the main character Ales is facing in the real world, a memory S is retrieved which shares some similarities with T to be used as the source scene. Then the scene T is continued based on the way the memory S unfolded. This process is divided in 4 steps:

- **Scene retrieval:** a suitable memory S is selected to be the source of analogy.
- **Mapping generation:** a mapping between the characters, objects, actions, locations, etc. of S and T is established.
- **Scene graph construction:** the graph representing the target scene T is extended by analogy with T , i.e. Riu generates the CuD of the generated story.
- **Text generation:** Riu extends the set of sentences representing T to generate the HuD of the generated story.

Notice, moreover, that even if *text generation* is listed here as a separate step in our process, it's more a *consequence* of the scene graph construction step than a separate step. Let us explain each one of these steps in more detail.

Memory Retrieval

Riu retrieves memories by looking for the most similar memory to the situation at hand. Similarity is evaluated using a two step process:

Surface Similarity: Riu first extracts a series of keywords from the current scene and every candidate memory, and selects the k memories with the most overlapping keywords with the current scene (In Riu, $k = 3$). Keywords are generated by considering the labels in all the nodes of the graph

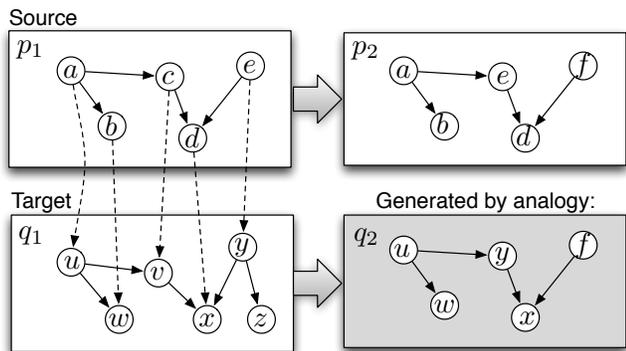


Figure 5: An illustration of the mapping found by SME among the nodes of the graphs in the CuDs of two scenes.

which represents the scene. For example, the keywords of the memory in Figure 2 are: happy, play, young, ales, have, bird, animal, sad, dead (i.e. the non-force-dynamics nodes in the graph).

Structural Similarity: Then, SME is used to compute analogical mappings and their strength, between each of k selected memories and the current scene. As indicated by the structural mapping theory, SME favors deeper (i.e., structural) similarity over surface (i.e., isolated nodes) similarity. Therefore, the memory that shares the largest structures with the current scene will receive the highest score from SME.

The rationale behind this two step process is that structural similarity is a computationally expensive process (specially with potentially large structures such as some of the scenes); thus, surface similarity is used to trim down the candidate memories to a small number. This is a well established procedure for analogy based memory retrieval systems such as MAC/FAC (Gentner and Forbus 1991).

Mapping Generation

Let us assume we have a source scene S composed of n phases and target scene T composed of m phases, where $m < n$. Riu computes a mapping between the first m phases of S and T using SME.

Specifically, Riu takes the nodes in the graph representing S which compose the first m phases, and computes a mapping using SME with the complete graph which represents T . The result is a mapping $M : S \rightarrow T$, which maps each node n from the graph representing S to a node $n' = M(n)$ from the graph representing T . If a particular node n has no matching node in T , then $M(n) = \perp$.

For instance, in the example shown in Figure 1 Riu builds a mapping between the memory of a pet bird Ales used to have and the current scene when Ales finds a cat in the street. One of the mappings found by SME is that $M(bird) = cat$.

Scene Graph Construction

After a mapping M is found using SME between the elements in the source scene and the target scene, Riu extends the shorter target scene graph with additional phases generated by analogy from the longer source scene, as illustrated in Figure 5, where a target scene with a single phase

is extended by adding an additional phase by analogy with a scene containing two phases. Let $P = \{p_1, \dots, p_m, \dots, p_n\}$ the n phases of the source scene S , and $Q = \{q_1, \dots, q_m\}$ the phases of the target scene T . Riu will generate additional phases $Q^* = \{q_{m+1}, \dots, q_n\}$ in T by analogy from S in the following way. For each new phase to generate $q_i \in Q^*$:

1. Let S_i be all the nodes in the graph representing phase p_i in S . Riu computes the set $S_i^M = \{n \in S_i | M(n) \neq \perp\}$ composed of all the nodes in the graph representing phase p_i that have been mapped to a node in the target scene. For example, in Figure 5, $S_2^M = \{a, b, e, d\}$.
2. The next step is to compute the set of nodes $S_i^{M^*} \in S_i$, corresponding to the set of nodes which are connected directly or indirectly to any node in S_i^M . $S_i^{M^*}$ corresponds to the nodes which are related in some way to some node which has some relation to the target scene. This is the set of nodes which will be transferred to T . For example, in Figure 5, $S_2^{M^*} = \{a, b, e, d, f\}$.
3. Finally, the new phase q_i consists of a copy of the graph, but only containing the nodes in $S_i^{M^*}$, and in which each node $n \in S_i^{M^*}$ such that $M(n) \neq \perp$ has been substituted by $M(n)$. For example, in Figure 5, notice that the nodes $\{a, b, e, d\}$ in the new phase q_2 , have been substituted by $\{u, w, y, x\}$.

Once this process is complete, Riu already has extended the computer-understandable description of the target scene by analogy from the source. The longer the number of phases the source scene, the longer the story generated in the target scene. Next section explains how text is generated to match the new phases $\{q_{m+1}, \dots, q_n\}$.

Text Generation

The final stage of the analogy-story generation process in Riu is the generation of the human-understandable description of the target scene. To perform this process for one of the newly generated phases q_i , Riu does the following:

1. First, Riu identifies the set of English sentences C_i from the source scene, associated with the phase p_i , and that are linked to any of the nodes in the graph of the newly generated phase q_i .
2. Now, for each node n in the graph representing p_i that was mapped to a node in the source phase q_i , Riu finds the pieces of sentences which correspond to $M(n)$. For instance, in Figure 4, the piece of English sentence which corresponds to the node *have* is “Ales used to have a bird”.
3. Finally, all the pieces of sentences in the sentences C_i which are linked to a node n , are substituted by the text corresponding to $M(n)$. For instance for the sentence “Ales used to have a bird when he was young”, since $M(bird) = cat$, and the text associated with the node *cat* is “cat”, the resulting sentence looks like: “Ales used to have a cat when he was young.”

After this process is over, Riu has generated both the HuD and CuD of the generated scene, thus completing the analogy-based story generation process.

<i>Source: M1</i>	Ales used to play with a bird when he was young. Ales played a lot with the bird and was very happy. But the bird died, leaving him really sad.	<i>Target: S1</i>	One day, Ales was walking in an alley. when he saw a cat in front of him. Ales hesitated for a second about what to do with the cat since he was late for work. > PLAY
<i>Result: R1</i>	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.		
<i>Source: M2</i>	Ales remembered the garage in which he had his first oil change, it was all red. His owners said he was rusty, and forced him to change his oil, he was a fool to accept.	<i>Target: S1</i>	One day, Ales was walking in an alley. when he saw a cat in front of him. Ales hesitated for a second about what to do with the cat since he was late for work. > PLAY
<i>Result: R2</i>	One day, Ales was walking in an alley. when he saw a cat in front of him. Ales hesitated for a second about what to do with the cat. Ales remembered the garage in which Ales had his first play with the cat, it was all red. work said he was rusty, and forced Ales to play with the cat, he was a fool to accept. Ales felt very awkward afterwards, and decided that Ales would have to be really rusty before his next play with the cat. How come no one ever complained about play with the cat?		
<i>Source: M3</i>	Ales had always wanted to be a painter. He was even learning to be a painter. But the long hours of his day job leave him very little time to practice. Eventually giving up. He stores different odd-looking objects and opes one day he can draw them.	<i>Target: S2</i>	Ahead, some incident occurred at the intersection. A big crowd of people blocks the way. > GO-BACK
<i>Result: R3</i>	Ahead, some incident occurred at the intersection. A big crowd of people blocks the way. Ales has always wanted to walk against the crowd. He was even walk against the crowd. But the long hours of his day job leave him very little time to walk against the crowd eventually giving up.		

Figure 6: Some examples of stories generated by analogy by Riu. Stories are always generated by continuing one target scene (right) with what happened in a source scene (left).

Experimental Results

In order to evaluate our technique, we asked Riu to generate stories by analogy, and generate English output text for each of them. Figure 6 shows three different stories generated by analogy by Riu. For each of the three stories, we show both the source (left) and the target scene (right).

The first generated story, R1, is generated by extending a situation when Ales finds a cat in the street (S1) and the user decides to play with the cat by analogy with a memory when Ales had a pet bird (M1). Among others, Riu maps the bird to the cat, and generates a story where Ales plays with the cat and was happy, but then the cat died, leaving him sad, which is how the memory of the bird ended. Notice that the output text is automatically generated by Riu.

The second generated story, R2, is generated by extending the same situation of the cat in the street (S1) with a memory of when Ales had his first oil change. In this case, Riu finds the mapping of “playing with the cat” to “getting an oil change”, and “work” to “his owners”. This last mapping is found because “work” and “his owners” are the antagonists in each story, in one scene he cannot play with the cat because he’s late for work, and in the other one he has to get an oil change because his owners want. This mapping can be found thanks to the force dynamics structure in the scene graph. Notice, moreover, that in the resulting story, there are more sentences than in the source scene. This is because

some of the sentences in the source scene were optional (according to the finite state machine which controls text generation), and didn’t get generated in the source scene, but did in the resulting story. Also, notice that there are some grammatical errors in the generated text, since Riu does not incorporate any information about verb tenses. Which is part of our future work. More importantly, there are also some semantic errors (e.g. “work said”), which occur since Riu does not have any further knowledge of what the words mean. As part of our future work, we would like to experiment adding additional semantic knowledge during the analogy process to improve these aspects.

The third generated story, R3, is generated by extending a situation when Ales finds a big crowd in the street (S2) by analogy with a memory where he remembers he wanted to be a painter but couldn’t because of his job (M3). In this case, Riu found the mapping between “learning to be a painter” with “walking against the crowd”.

As we can see, Riu’s analogy-based story generation successfully generates stories, including text descriptions of them, avoiding mechanically generated text, typical of planning-based story generation systems. Moreover, we have seen how force dynamics can be used to find deep analogical mappings, like “getting an oil change” with “playing with the cat”, which do not seem to share any surface similarity, but that play the same role in the scenes.

Related Work

Although planning is one of the most common techniques for story generation, there have been a number of systems that use other techniques like case-based reasoning. A thorough overview can be found in (Zhu and Ontañón 2010b).

Among the systems that adopt classic computational analogy, Riedl and León's system (Riedl and León 2009) combines analogy and planning. It uses analogy as the main generative technique and uses planning to fill in the gaps in the analogy-generated content. The system uses the CAB computational analogy algorithm (Larkey and Love 2003) for story generation and uses a representation consisting of planning operators. This system, however, focuses only on content generation without paying attention to text generation. The PRINCE system (Hervás et al. 2007) uses analogy to generate metaphors and enrich the story by explaining a story existent in the domain T using its equivalent in S.

The human-understandable description used in Riu is inspired by GRIOT (Harrell 2007), which uses Harrell's ALLOY conceptual blending algorithm to produce affective blends in the generated poetry (GRIOT) and narrative text (MRM). GRIOT contains a computer-understandable description of some of the terms in its sentence templates, which are manipulated by ALLOY, in a similar way to Riu. The main difference is that in GRIOT generativity is at the word level, i.e. GRIOT uses conceptual blending to generate descriptions, like adjectives or combinations of name plus adjective, while in Riu we are generating complete stories.

Other systems such as Minstrel (Turner 1993), use case-based reasoning (CBR). These CBR systems also use analogy-like operations to establish mappings between the source cases and the target problem. Reminiscent of CBR, MEXICA (Pérez y Pérez and Sharples 2001) generates stories using an engagement/reflection cycle. It maps the current state to the states in the pre-authored memories and retrieves the most similar one for the next action. The library of past memories in Riu can be seen as the case base of authored stories in the CBR-based systems.

Conclusions

This paper focuses on analogy-based story generation and in particular on two important aspects: a) how to represent stories in a way that is amenable for computational analogy methods, and b) how to link the process of computationally generating the content of the story and how to generate text representing the story. We have implemented our approach in the Riu system.

Computational analogy is a promising approach to story generation. Compared to planning-based approaches, analogy-based story generation offers a less goal-oriented story generation process. However, planning-based techniques allow the specification of a goal state, and thus have a control of the direction the generated story should take. Moreover, as we have shown in this paper, carefully designing the story representation allows the system to manipulate the computer-understandable description, and as a result be able to generate a human-understandable representation of the generated story. Thus, achieving text generation as a

consequence of the story generation process, rather than as a post process. The work presented in this paper is a first step towards bringing attention to the finally generated text when using AI story generation techniques.

As part of our future work we want to explore ways in which the system can better understand and manipulated the human-understandable description than by mere substitutions, for instance allowing verb tense manipulations. We also plan to explore the possibilities of our method with larger and more complex stories. Additionally, we would like to perform user studies to assess the perceived quality of the stories generated by analogy.

References

- Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41:1–63.
- Gentner, D., and Forbus, K. D. 1991. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science* 19:141–205.
- Harrell, D. F. 2007. *Theory and Technology for Computational Narrative: An Approach to Generative and Interactive Narrative with Bases in Algebraic Semiotics and Cognitive Linguistics*. Dissertation, University of California, San Diego.
- Hervás, R.; Costa, R. P.; Costa, H.; Gervás, P.; and Pereira, F. C. 2007. Enrichment of automatically generated texts using metaphor. In *MICAI*, 944–954.
- Larkey, L. B., and Love, B. C. 2003. Cab: Connectionist analogy builder. *Cognitive Science* 27(5):781–794.
- Murray, J. H. 1998. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Cambridge: The MIT Press.
- Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *J. Exp. Theor. Artif. Intell.* 13(2):119–139.
- Riedl, M., and León, C. 2009. Generating story analogues. In *AIIDE 2009*. The AAAI Press.
- Talmy, L. 1988. Force dynamics in language and cognition. *Cognitive Science* 12(1):49–100.
- Turner, S. R. 1993. *Minstrel: a computer model of creativity and storytelling*. Ph.D. Dissertation, University of California at Los Angeles, Los Angeles, CA, USA.
- Woolf, V. 2002 (1925). *Mrs. Dalloway*. Harcourt.
- Zhu, J., and Ontañón, S. 2010a. Story representation in analogy-based story generation in riu. In *IEEE-CIG*, to appear.
- Zhu, J., and Ontañón, S. 2010b. Towards analogy-based story generation. In *First International Conference on Computational Creativity (ICCC-X)*.
- Zhu, J. 2009. *Intentional Systems and the Artificial Intelligence (AI) Hermeneutic Network: Agency and Intentionality in Expressive Computational Systems*. Ph.D. Dissertation, Georgia Institute of Technology.