Say Anything: Using Textual Case-Based Reasoning to Enable Open-Domain Interactive Storytelling

REID SWANSON, The University of California Santa Cruz
ANDREW S. GORDON, The University of Southern California

We describe Say Anything, a new interactive storytelling system that collaboratively writes textual narratives with human users. Unlike previous attempts, this interactive storytelling system places no restrictions on the content or direction of the user's contribution to the emerging storyline. In response to these contributions, the computer continues the storyline with narration that is both coherent and entertaining. This capacity for open-domain interactive storytelling is enabled by an extremely large repository of nonfiction personal stories, which is used as a knowledge base in a case-based reasoning architecture. In this article, we describe the three main components of our case-based reasoning approach: a million-item corpus of personal stories mined from internet weblogs, a case retrieval strategy that is optimized for narrative coherence, and an adaptation strategy that ensures that repurposed sentences from the case base are appropriate for the user's emerging fiction. We describe a series of evaluations of the system's ability to produce coherent and entertaining stories, and we compare these narratives with single-author stories posted to internet weblogs.

Categories and Subject Descriptors: I.2.1 [**Artificial Intelligence**]: Applications and Expert System—*Games, Natural language interfaces*

General Terms: Human Factors

Additional Key Words and Phrases: Interactive Storytelling, case-based reasoning, weblogs and social media

ACM Reference Format:
Swanson, R. and Gordon, A. S. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. ACM Trans. Interact. Intell. Syst. 2, 3, Article 16 (September 2012), 35 pages.
DOI = 10.1145/2362394.2362398 http://doi.acm.org/10.1145/2362394.2362398

1. MOTIVATION

Digital interactive storytelling is a compelling new medium for expressing and communicating ideas where a normally passive storytelling experience is transformed into an active engagement in the creative process. Recent years have seen incredible progress in graphics and physical modeling, enabling the creation of visually stunning interactive virtual environments. The attraction to these visual capabilities has helped spark the imagination of an ever-widening audience and fueled the desire for complex

The reviewing of this article was managed by the special issue associate editors, Henry Lieberman and Catherine Havasi.
This work was sponsored by the US Army. The content or information presented does not necessarily reflect the position or policy of the US Government and no official endorsement should be inferred.

Authors' addresses: R. Swanson (corresponding author), Computer Science Department, University of California Santa Cruz; email: reid@soe.ucsc.edu; A. S. Gordon, Institute for Creative Technologies, University of Southern California.

DOI 10.1145/2362394.2362398 http://doi.acm.org/10.1145/2362394.2362398

narrative interactions that parallel the richness of the audiovisual experience. However, progress on the narrative side of interactive experiences has advanced much more slowly. One of the primary reasons for the difficulty in this area is the incredible amount of common sense knowledge required about people's daily lives and activities that is required in order to reason about narrative. This problem is typically solved using hand-authored domain theories, but so far these approaches have been unable to scale beyond small toy domains. In this article, we describe a new data-driven approach that leverages millions of stories from blog entries to enable a real-time interactive narrative system that breaks from existing paradigms.

Majewski [2003] argues that there are five possible models for introducing narrative into nonlinear environments. First is the *linear narrative* model, which simply removes the choice from the user and forces the prescribed narrative onto the experience. Second is the *string of pearls* model, a popular approach in video games, which strings together a sequence of mini-episodes. Within a single mini-episode the player has some freedom to explore and uncover optional game elements, but ultimately they must achieve a specific linear goal in order to proceed to the next episode. Third, the *branching* model presents the user with multiple-choice decisions at various points in the experience, which are used to select a path within a pre-authored narrative tree. Whereas the branching model creates narrative complexity over time, the fourth model, the *amusement park*, distributes this complexity over space, allowing the player to access different narrative strands by exploring the game world. Fifth, the *building blocks* model eschews explicit narrative altogether, providing instead a rich environment for user-directed action in which the player is the one making all of the important decisions and who must ultimately interpret these experiences as a coherent narrative in their own mind.

Majewski's categorization defines a spectrum of interactive narrative models that place different burdens on the authors of these systems. On one end of the spectrum, system authors employing the linear narrative model have full control over the story (the user has none) and can explore the full range of genres, topics, and storytelling devices that have been perfected in other narrative media. On the opposite end, system authors employing the building blocks model enable user-directed action by designing a world governed by generative models, that is, the "physics" of the world that provides the users with action effects that are coherent and entertaining. Authors in these systems are modelers of world behavior, and the authorial control in these systems is in the design of how the world works. In this approach, interactivity is enabled when user-directed action intersects with the domains of the modeled environment. Unsurprisingly, the most successful examples of the building blocks approach enable interactivity in domains that are relatively easy to model and visualize, for example, the physical interaction between solid objects. In domains that lack suitable models altogether, for instance, the social interaction among people, this model of narrative interactivity often falls flat [Milam et al. 2009].

Because narrative is so deeply connected to the social interactions and experiences of people, the prospect for open-domain user-directed interactive storytelling is highly contingent on the development of generative models across the full spectrum of narrative domains. Individual domain models of this sort have a long research history in artificial intelligence where logical formalizations have been developed for various phenomena of physics [Hayes 1985], sociology [Hobbs and Sagae 2011], and psychology [Gordon and Hobbs 2004]. However, few researchers in this area today believe that robust computational theories across the full spectrum of narrative domains are just around the corner. For the foreseeable future, open-domain interactive storytelling will need to rely on other approaches that produce generative models of what should happen in the story.

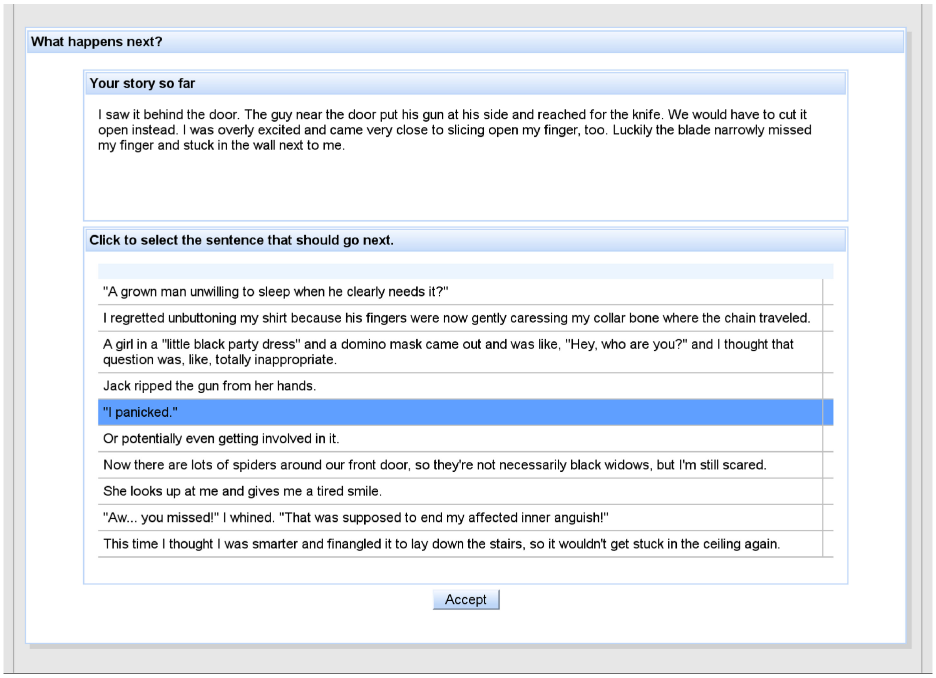
One attractive alternative is case-based reasoning. Broadly speaking, the approach of case-based reasoning is to trade off a deep theoretical understanding of a domain for lots of experience, relying on a rich memory of past solutions to solve current novel problems. Research on case-based reasoning has its own rich history in artificial intelligence and has taken on creative challenges from Schechwan cooking [Hammond 1989] to the expressive performance of jazz ballads on the saxophone [Arcos et al. 1998]. The applicability of case-based reasoning to open-domain interactive storytelling requires a reconsideration of the notions of problems and solutions. Here the problem is: What happens next? given the user's current choices and what has happened so far in the user experience. Solutions, in turn, answer the question of What happened in previous, similar situations? From an engineering perspective, the success of this approach requires a means of amassing of suitably large case library and the development of competent retrieval and adaptation mechanisms. There are several routes that could be taken to overcome these engineering challenges. Here we explore a textual case-based reasoning approach [Weber et al. 2006] where case retrieval and adaptation are processes that operate on textual representations of cases. By reducing the complexity of case representation as compared to previous case-based reasoning applications, we focus on the fundamental problem of scale, namely, developing case libraries that spanned the full breadth of narrative domains.

In this article we describe an application called *Say Anything*, which addresses the problem of open-domain user-directed interactive storytelling using a textual case-based reasoning approach. Instead of relying on domain knowledge acquired by domain experts versed in specific formal languages, this system leverages the personal stories posted by millions of people to their Internet weblogs. *Say Anything* is a text-based interactive storytelling application in which the human and computer take turns writing sentences of an emerging story. By using weblog stories as a knowledge base, this system is able to interpret the story that has been written so far and to generate a sentence that continues the narrative in an appropriate way, regardless of the narrative choices made in the user's contribution.

The following paragraph is an example of the type of story domain and inference we can support, which would be difficult in other frameworks.

> You'll never believe what happened last night! *Leigh laughed at my joke but I couldn't help but think 'liz would have laughed harder.'* The joke wasn't very funny in a "ha ha" kind of way. *It wasn't anything like that, I thought he was going to give me a good night kiss but he ended up licking my cheeck, she declared.* It made me sneeze and snort out loud. *And now my nose hurts from the snorting.*

In this story, every other sentence (in italic) was produced by the computer during a writing session with the system. Although this is not the same quality one would expect from an experienced novelist, it illustrates several beneficial properties of our system. First, the narrative is about the lives of people and the experiences they find funny and unexpected. To reason effectively in this domain a large amount of world knowledge is needed, including the types of activities people typically engage in, what they find unexpected, and even the types of biological responses given certain physical stimuli. For example, it correctly concludes that a joke does not have to be verbal, but can be an unexpected physical act, such as licking a cheek. It also makes a remarkable inference that snorting is a type of physical act particular to a nose and that it can actually hurt if it is done with enough force. While these types of inferences could be implemented in standard domain engineering approaches, many of these mundane



I saw it behind the door. The guy near the door put his gun at his side and reached for the knife. We would have to cut it open instead. I was overly excited and came very close to slicing open my finger, too. Luckily the blade narrowly missed my finger and stuck in the wall next to me.

"A grown man unwilling to sleep when he clearly needs it?"

I regretted unbuttoning my shirt because his fingers were now gently caressing my collar bone where the chain traveled.

A girl in a "little black party dress" and a domino mask came out and was like, "Hey, who are you?" and I thought that question was, like, totally inappropriate.

Jack ripped the gun from her hands.

"I panicked."

Or potentially even getting involved in it.

Now there are lots of spiders around our front door, so they're not necessarily black widows, but I'm still scared.

She looks up at me and gives me a tired smile.

"Aw... you missed!" I whined. "That was supposed to end my affected inner anguish!"

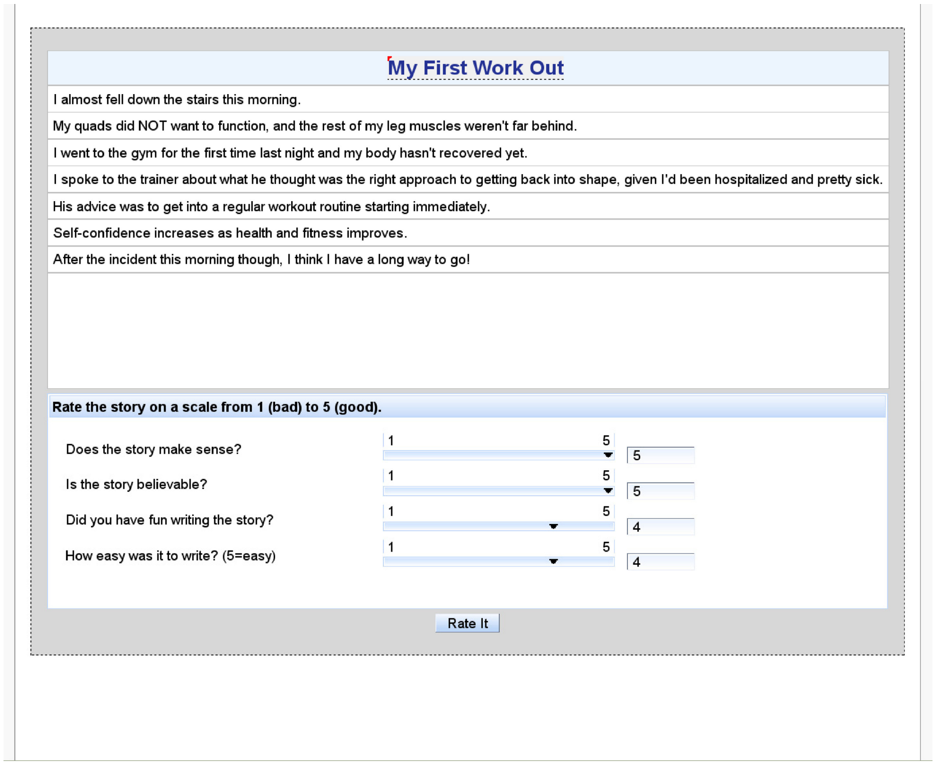
This time I thought I was smarter and finangled it to lay down the stairs, so it wouldn't get stuck in the ceiling again.

Fig. 1. Selection dialog window.

pieces of information are usually excluded because of the immense cost and difficulty of authoring.

2. SAY ANYTHING OVERVIEW

The goal of *Say Anything* is to collaboratively write a story with a human user that is entertaining and coherent regardless of the topic or direction of the user's contributions. We break from previous interactive storytelling designs by pursuing a loosely structured data-driven approach that leverages the millions of stories authored by ordinary people on their weblogs about their everyday life experiences. In this application, a user is initially presented with an empty story. The collaborative process begins when the user enters the first sentence of the narrative. To add the next sentence, the system analyzes the story as it exists so far, and finds a similar set of instances in a massive knowledge base of over one million stories. Using these retrieved stories as solutions to the problem of what happens next, the system displays the corresponding next sentences from these stories, offering each as a choice to the user for how the narrative will continue. An example of this interaction is shown in Figure 1 where the system displays ten sentences from ten different stories that are similar to the user's story as it exists so far. After selecting one of the options, the user can write another sentence and the cycle continues until the user decides the story is complete or too far off track. When the user is finished with her story, she is asked to rate the story on several criteria, shown in Figure 2. In the remainder of this section, we will provide a more detailed example that illustrates how stories are created in our system. We then outline the three major system components that are discussed in the remainder of this article.



My First Work Out

I almost fell down the stairs this morning.

My quads did NOT want to function, and the rest of my leg muscles weren't far behind.

I went to the gym for the first time last night and my body hasn't recovered yet.

I spoke to the trainer about what he thought was the right approach to getting back into shape, given I'd been hospitalized and pretty sick.

His advice was to get into a regular workout routine starting immediately.

Self-confidence increases as health and fitness improves.

After the incident this morning though, I think I have a long way to go!

Rate the story on a scale from 1 (bad) to 5 (good).

Does the story make sense?	1 … 5	5
Is the story believable?	1 … 5	5
Did you have fun writing the story?	1 … 5	4
How easy was it to write? (5=easy)	1 … 5	4

Fig. 2. Rating interface.

2.1. Example

Figure 3 illustrates how the *Say Anything* system identifies candidate sentences to contribute to the user's story. On the left-hand side is a full story, authored using *Say Anything*, where every other sentence (in *italics*) has been contributed by the system. Here we focus on two sentences added late in the story, which appear between the horizontal lines, representing one turn of the system. The user wrote the first of these two sentences, while the second was contributed by the system. To suggest this sentence as a plausible continuation of the user's story, the system first searches its case base for stories that are similar to the user's story as it exists so far. On the right-hand side of the example is the corresponding portion of the weblog story determined to be the best match for the user's story when this turn was taken. The matching algorithm pays particular attention to the similarity of the last sentence to one of the sentences in the retrieved story where the overlapping words of "pay," "repairs," and "car" help establish a correspondence between the two texts. From this correspondence, the system selects the next sentence in the retrieved story as a plausible continuation of the user's story: "Where are we going to get the money?"

The user's story in this example demonstrates that *Say Anything* can be used to write relatively long stories that maintain a moderate degree of coherence. It would be wrong to claim that this story exemplifies deep aspects of modern narrative theory, but some basic narrative structure is still readily evident. It begins with a generality that introduces the subject of the story and contextualizes the specific narrative events that follow. It continues with some background information, such as when the events take place and that the main character is married, setting the scene for the remainder

User's Story	Retrieved Story
Untitled	... continued above ...
Even the best laid plans go awry. *I knew when the moment arrived and it happened the day before we were flying.* Snowstorms were forcast throughout the entire midwest. *I started the morning out by dropping my wife off to work.* On the way homne, I got a flat tire. *The irony is that I remember thinking that morning when I got in the car. "Man I wish we had a 3day weekend coming up soon, I could use a day off to get some new tires and maintenance done on the car."* A four hour wait fro roadside assistance caused me to miss my lunch meeting. *By the time I was done with my dental cleaning, the repair serviceman had already finished the work, and just needed me to sign the paperwork.* When I tried to pay my dental bill, my credit card was declined. *This was going from bad to worse.*	All through this, I cried 2 more times. And again when we get to the body shop. Agnes, who owns the place, told my mom what might happen in regards to the insurance situation. It's looking a bit grim. I don't understand the ins and outs of insurance policies, but I'm pretty sure that because I'm not insured, the insurance company is probably not going to let my mom make any claims. And even if they do, it'll probably be less or something. I'm not sure. My mom hasn't been able to get a hold of her agent, yet. The police report is going to take a few days, but I'm going to call the station tomorrow to see if they might have gotten around to it. The biggest problem is money. My mom has such a low income nowadays that I don't think she can afford to rent a car and wait until the car is fixed.
Now I had no way to pay for the repairs to my car or the dentist. *Where are we going to get the money?*	Then she would have to pay for the repairs ($3000-$4000) AND the car rental. Where are we going to get the money?
Plus our vacation starts tomorrow with no way to pay for expenses. *My 2 weeks in tjeschie were extremely boring, because there was nothing to do, except visiting castles.* I should have known as soon as I got that flat tire that things just weren't going smoothly.	

Fig. 3. The left column shows an entire user story. The right column looks at part of the weblog story used as a proxy for generating the highlighted sentence. The words highlighted in dark gray show the overlap leading to the highest similarity score.

of the story. Most of the remainder of the story describes the events constituting the plot of the narrative.

The plot itself is primarily a sequence of unfortunate events that does very little to advance the story to a climactic moment. The user does something clever with the last sentence that enables a basic narrative arc to be constructed despite the relative lack of structural elements previously in the story. Essentially, they provide a moral to the story that explains why all of the other discourse was necessary. It is a common device used in both the collection of weblog stories and in the user-generated stories created during these experiments. In this case, it seems to work quite effectively by providing a clear resolution as well as enabling the second to last sentence to function as a climax or falling action. A simple interpretation of the complete narrative arc is illustrated in Figure 4.

Although the user's story is successful on many levels, this example also illustrates several problems that are the focus of subsequent sections of this article. One of the biggest areas of concern is the tendency of these interactive stories to wander off topic after only a few turns. This is a concern shared with other sequential problems that must make a Markovian assumption to maintain tractability. We have not completely

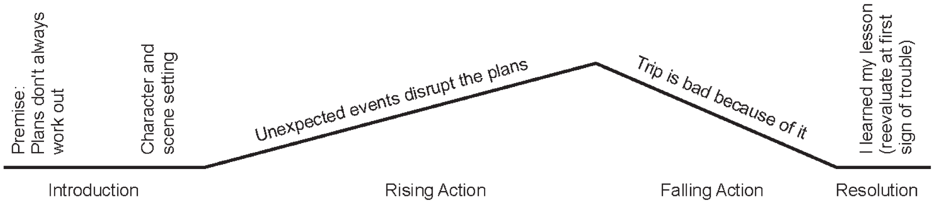

Premise: Plans don't always work out
Character and scene setting
Unexpected events disrupt the plans
Trip is bad because of it
I learned my lesson (reevaluate at first sign of trouble)
Introduction
Rising Action
Falling Action
Resolution

Fig. 4. Simple narrative arc of the user's story in Figure 3.

solved this problem, but we will show in Section 4 how to make use of long-term dependencies that will greatly improve the situation.

2.2. System Architecture

The architecture of our system is a remarkably simple design that follows the basic philosophy of case-based reasoning (CBR). In this type of reasoning, new problems are solved analogically by reusing and adapting solutions to problems that have already been seen before. In order to support this type of reasoning there are four necessary components.

(1) A case library of previously solved problems;
(2) A retrieval mechanism for matching current problems to cases in the library;
(3) An adaptation component for modifying previous solutions to the unique properties of the current problem;
(4) The ability to reuse knowledge learned as the system solves new problems.

This design is also consistent with emerging conceptions of textual case-based reasoning [Weber et al. 2006]. Textual case-based reasoning follows the same high-level philosophy as traditional CBR, but tries to reason directly from cases represented as natural language text. Working with natural language introduces several specific challenges, such as rethinking optimal similarity measures, how to map free-form text to structured representations, and what types of adaptation are possible when using this type of data. The payoff is a system that has access to vast amounts of common sense knowledge, efficient real-time reasoning, and authentic sounding language generation.

Following the CBR design, our system has three main components: a *case library* of previously solved problems, a *retrieval* mechanism for finding similar cases in the library, and an *adaptation* component that fits old solutions to new problems. In Section 3, we describe how we obtained our case library, which consists of over one million stories told by ordinary people, describing their everyday life experiences. These linear stories are then treated as though they were perfect solutions to our collaborative story writing problem. When the user is finished contributing their sentence, we analyze the user's story to try to find the most appropriate weblog story from our collection. In Section 4, we discuss how the retrieved stories are used to generate responses for the user's narrative, and we also describe how we map the raw text of a story to several different structured representations. In this section, we also show how the selection history of every user can be reincorporated into the system to improve the candidates proposed by the system in the future. In Section 5 we discuss the adaptation process that alters retrieved sentences to better match the new story being written.

3. CASE LIBRARY: BUILDING A CORPUS OF PERSONAL STORIES

Although the full space of possible narratives is infinite, we take the position that the number of narrative schemas that describe their underlying activities and events is a much smaller set. Even though any new story written with our system will be unique in its prose and discourse, it is very likely that someone, somewhere has already written a story following the same general narrative schema. In our framework we generate a narrative through the interpretation of the current story in relation to an instance already existing in our corpus and a projection from the existing discourse to the emerging narrative. By posing the generation problem in this way we can treat any linear narrative as a solution to a previously solved case. Therefore, its success ultimately hinges on the availability of a large-scale corpus of narrative stories that has sufficient coverage over a broad range of human activities. Furthermore, this corpus should also have the depth to produce the type of sophisticated and nuanced variations that make human-authored stories so captivating.

Collections and anthologies of stories are as old as the first written stories themselves. Most anthologies consist of only a few dozen or a hundred stories. One of the first efforts to collect a large-scale corpus of stories was the Federal Writer's Project of the Works Project Administration [Mangione 1996], which solicited thousands of personal and fictional stories capturing the life and culture of the time as part of a broad economic stimulus package started in the 1930s. StoryCorps[1] is a contemporary nonprofit organization modeling itself after the Federal Writer's Project to record and share the personal stories of Americans from all types of backgrounds. It is one of the largest manual collections of narrative experiences with over 30,000 recorded interviews.

Despite the advantages of manual story collection strategies, this process also has a critical drawback for our application. On the positive side, it is much easier to target specific domains or create rich ontological categorizations based on topic, demographic, and other criteria. On the down side, these collections only describe the stories and events that people feel are important enough to share with a special service. As humans, we are able to process these stories because we possess the background knowledge to read between the lines and understand what makes them so special. However, it is precisely this background knowledge that is largely missing from these collections, which enable reasoning about narratives at all. For this purpose StoryCorps and other manual collections of “interesting” stories are far too small to contain the breadth and depth of common sense knowledge necessary to enable the type of open-domain reasoning we hope to accomplish. To capture this knowledge we are equally interested in the “boring” stories people tell about everyday life. These contain the simple inferences needed for narrative coherence, such as what happens when you forget your spouse's birthday or lock your keys in your car.

Everyday life stories have existed in the text of personal diaries for centuries [Steinitz 1997]. Until recently, these documentations of mundane aspects of people's lives have been largely unavailable to anthropologists, historians, and computer scientists interested in the intricate knowledge about human society and behavior they possess.

Fortunately, the World Wide Web has enabled people to effortlessly share this type of story with the entire world through weblogs and other forms of social media. Blogging was introduced to the world in the late 1990s, but was initially embraced only by those immersed in information technology. A few years later, there was a dramatic explosion in user-generated content, mainly credited to the emerging content-creation tools allowing people to publish their thoughts with little technical ability [Blood 2000]. In 2008, Technorati.com estimated that nearly one-million weblog entries were posted to the Web every day [Technorati 2008].

[1] http://storycorps.org/

Despite the huge amount of user-generated content, most weblog entries are not journal-style personal reflections on daily or weekly events [Blood 2000], which is our primary interest. Many weblogs are filter-style, acting as portals to existing content where users comment and post opinions about world events, politics, sports, and other issues that are relevant to the target audience. Weblogs have also become a popular platform for many businesses to publish product releases, company newsletters, daily specials, advertisements, and spam. The remainder of this section will describe the methods used to find and extract the journal-style weblog entries containing personal stories about the author's own experiences in order to create our case library of personal narratives.

3.1. Annotating Blog Entries as Stories

The goal of our collection effort was to obtain the broadest collection of personal stories about everyday life experiences. The Web is an ideal place to find massive amounts of user-generated textual content, however, even with automated tools such as Nutch[2], identifying and crawling weblogs is a nontrivial task. Fortunately, social media has become such a high-demand commodity that it has spawned a new community of intermediate service providers that sort, filter and compile this type of content, including blog entries, into a more convenient and accessible package.

One of these companies, Spinn3r.com, released 44 million weblog entries spanning a two-month period as part of the 2009 International Conference on Weblogs and Social Media. In addition to simply crawling the sites and providing the raw content, Spinn3r.com also includes metadata associated with each post provided by Web feed information, such as Atom Syndication and RSS. This information includes things like the name of the author, the time and date the post was published, and any category tags the author used to describe the post. Along with the syndicated metadata, Spinn3r.com also includes information from several custom applications that attempt to identify spam, infer the language, and extract the main content of the post.

Following the work of Gordon and Ganesan [2005], who first proposed the task of identifying stories from text, we treated the problem as a binary classification task. To develop a gold-standard training corpus, we randomly selected 5,270 English weblog posts from the Spinn3r.com dataset. Each of these entries was hand labeled by two annotators according to the definition proposed by Gordon and Ganesan [2005] with the following clarification.

> Discourse that describes a specific series of causally related events in the past, spanning a period of time of minutes, hours, or days, where the storyteller or a close associate is among the participants.

Annotating the corpus was done in an iterative process until all the examples were given a categorization that both annotators could agree upon. The first round of annotations began with the two annotators independently labeling each entry using the updated definition. In cases where a blog entry contained more than one distinct subject, it was only labeled a positive example if more than 50% of the content adhered to our definition. We used Cohen's κ to assess the level of agreement between the raters, which was found to be 0.68 at this stage. This indicates a relatively high level of agreement and is in line with the previous experiments by Gordon and Ganesan [2005]. After this process, 203 of the entries were classified as stories and there were still 177 disagreements.

[2] http://nutch.apache.org/

After completing the first round of annotation, the two annotators discussed in general terms, without looking at or referencing actual blog entries, some of the problematic issues that were encountered. In light of this discussion, each of the blog entries where there were reshuffled and classified again by both annotators. Once complete, κ was recomputed, which improved to 0.867. The number was disagreement of entries classified as stories increased to 229 and only 66 disagreements remained.

To resolve the final differences each of the remaining 66 entries were openly discussed between the annotators until they settled on a final judgment. Of these entries, there were three prominent reasons for disagreement. The first reason arose due to an ambiguity in the causal structure. Pure lists of events chronicling a person's experiences, which occur quite frequently, do not satisfy the given definition of a story because they lack the necessary causal structure. There is a continuous gradation between a simple chronology and a full narration, which leads to unpredictable classification between annotators. The second source of disagreement arose from the intent of the blog author. Typically the positive story examples that were initially agreed upon were authored specifically for the purpose of telling a story. However, many of the conflicting examples were a mixture of two distinct intents. For example, a cooking recipe might be described by telling the story of the first time the author prepared the dish. Another common example of this type of disagreement is technical help posts, for example, a post on an automotive repair site telling a story of how the author's car broke down. Although the purpose of these posts is to share (or ask for) knowledge and not explicitly to tell a story, it was decided to treat them as positive examples nonetheless. The third source of disagreement was based on the time scale of the story. A few entries closely followed our definition, but occurred over several years or even the entire life of a person. While not strictly adhering to our definition, a judgment was made on a case-by-case basis.

At the end of this process the gold-standard corpus contained 4,985 entries[3] of which 267 were annotated as stories.

3.2. Training and Applying a Story Classification Algorithm

Robust document classification can be a challenging task that requires a deep understanding of discourse structure as well as human cognition. For example, identifying sarcastic remarks not only requires the ability to parse the surface meaning of the utterance, but also to know enough about people's belief structures to know that the utterance is not to be taken literally. Automated text classification is a very active area of research[4] with a wide variety of techniques, including supervised, unsupervised, and semisupervised machine learning methods. For this work we chose a supervised machine learning approach, because these methods tend to perform best when the categories are predefined and there is sufficient training data.

Several of the features investigated in this section use syntactic parse trees labeled with dependency relations. In order to obtain this information, we developed a high-performance sentence detection algorithm that was applied to each entry. Our algorithm is a supervised classification-based approach similar to other state-of-the-art techniques such as mxTerminator [Reynar and Ratnaparkhi 1997] and Splitta [Gillick 2009]. Our approach is specifically designed for Web text and makes use of the HTML formatting to help inform boundary detection, which improved the F_1 performance by over 10% (precision: 0.943, recall: 0.897, F_1: 0.908) compared with the other

[3] The annotators disregarded any entry from the initial set of 5,270 if it was no longer available on the Web.
[4] The Bibliography of Automated Text Categorization (http://liinwww.ira.uka.de/bibliography/Ai/automated.text.categorization.html) maintains over 650 references up to 2007.

approaches. The resulting sentences were then submitted to an off-the-shelf shift-reduce dependency parser [Sagae and Lavie 2006] that is among the fastest and most accurate available.

Using the dependency information from the parse tree, a range of features were constructed to model each document. These included standard lexical bag-of-word (or *n*-gram) distributions that could capture the relative frequency of narrative indicators such as past-tense verbs, first-person pronouns, and temporal words, such as today, yesterday, week, etc. To capture slightly longer distance relationships in the text, we used the dependency relations obtained from the syntactic parses to construct dependency triples consisting of a verb and its subjects and objects. For example, the features *I:SUBJ:helped* and *Jane:OBJ:helped* would be extracted from the sentence, “I helped Jane last weekend”. We also considered a set of features that uses the distribution of syntactic roles of words across sentences, first introduced by Barzilay and Lapata [2005] for modeling document coherence.

In order to find the best features for automatically labeling the Spinn3r.com dataset, we split the gold-standard corpus just described into a development, training, and test set to assess the performance of our feature sets. The development set consisted of 250 weblog entries, the training set consisted of 3,985, and the test set contained 750. All engineering and debugging was performed on the training and development sets until all the feature sets were completely worked out.

Support vector machines and maximum entropy are common classifiers used in these types of natural language classification applications. Recently, several online algorithms have been proposed that rival the performance of these popular models, but are extremely efficient to train and apply. In particular, we chose to use a Confidence-Weighted Linear Classifier [Dredze et al. 2008]. This classifier is extremely efficient to train (orders of magnitude faster than SVMs), yet has been shown to perform equally or better in several NLP-related tasks. This classifier also has very few hyperparameters to set, so there is little or no search required for obtaining optimal performance.

We were able to obtain the highest accuracy using a combination of all our features reaching a precision, recall, and F_1-score of 0.591, 0.414, and 0.487. Similar to many other document classification tasks, trigrams were among the most predictive features and reached nearly the same levels of accuracy. These results are sufficient for our work, but in the future we believe there are still several ways to improve the precision and recall, such as better feature generation [Wiratunga et al. 2005] and selection [Guyon and Elisseeff 2003]. However, we believe increasing the size of the training data would provide the most improvement.

Using a model trained on all of the annotated data available, the story identification process and preprocessing steps described above were applied to the 25-million English language weblogs present in the Spinn3r.com corpus. 6.03% of the entries were extracted, resulting in 1,605,480 identified stories with an average length of 26.24 sentences. Of the 1.6-million stories, 80,142 were held out for development purposes and as independent training data for various type models that will be described in Section 4. After excluding these stories, 1,525,338 stories were included in the final dataset used by the system. A final postprocessing step was performed to clean up the data, such as capitalizing the first word of a sentence, removing repeated characters, and attempting to balance dangling parentheticals and quotations.

4. RETRIEVAL

Retrieval is the core inferential mechanism in case-based reasoning. It is the process of identifying a previously solved solution in our case library to enable reasoning about the current problem. This is generally a two-step process in which cases are first assigned indexes that provide an efficient method for search and retrieval, based on a desired similarity measure. Second, this index is used during interaction in order to query the database for candidate cases. In this section, we will describe the retrieval process for *Say Anything*.

Determining what constitutes similarity between textual cases has received a good deal of attention. Ultimately, the choice of similarity measure for a system is dependent on the structure and final representation of the case library. The vector space model is a common method used by many researchers that requires very little additional processing and often performs well in practice even when more information is available [Recio-García et al. 2007]. Similarity based on deeper semantic features has been shown to improve results in some cases [Burke et al. 1997; Mott et al. 2005], but these approaches can be difficult to scale to large datasets when real-time search is a requirement. For a detailed taxonomy of similarity metrics used in CBR, we refer the reader to Cunningham [2009].

In our application, an ideal characterization of these causal and temporal relations would be discriminative enough to operate on specific events mentioned in the narrative. Discourse parsing at this level is an important problem in the natural language processing community and a great deal of effort has been made to develop automated tools. Some of the most promising work has been the development of several annotated corpora, such as the Rhetorical Structure Theory Discourse Treebank [Carlson et al. 2001], the Penn Discourse Treebank [Miltsakaki et al. 2004] and TimeBank [Pustejovsky et al. 2003], which use different linguistic theories to try to capture these types of relationships. Despite the success of automating several complex linguistic tasks such as syntactic parsing, the performance of automatically extracting discourse structure is substantially lower than other processes in standard NLP pipelines, such as part-of-speech tagging and syntactic parsing. This is at least partly due to lower agreement between annotators, but also because the relationships often occur at a much greater distance or are not explicitly stated in the text at all. To compound the problem, only a limited amount of annotated data is available, and it is almost entirely news genre text. During a preliminary study, we found that a discourse parser trained on an existing news genre corpus suffered a dramatic loss in accuracy when applied to weblog data. Given the relatively low performance on the in-domain data, the results on the out-of-domain Web data were not acceptable for this work.

In the remainder of this section, we will discuss how we map our purely textual story cases into several structured representations, and how we compute similarity between these representations. In one sense, a case in our application is a single utterance (i.e., a sentence) since that is the unit of discourse that we use to interface with our user. On the other hand, we are also concerned with the stories that utterances compose because the discourse relations are what will allow for the appropriate narrative reasoning and enable coherent sentences to be projected. In general, we treat sentences as the basic unit of indexing and retrieval. However, every sentence maintains enough information to recreate the entire story, which can be used for additional processing.

Our approach relies heavily on information retrieval techniques, so we will begin with a brief review of the basic concepts in this area. We will then describe a simple complete story generation algorithm based solely on a standard vector space model. Finally, we will describe several new structured representations of our narratives and how we are able to use these representations to improve the reasoning process while maintaining real-time interaction.

4.1. Information Retrieval Infrastructure

Finding relevant passages of text is the foundation of information retrieval and is a common method used in textual case-based reasoning. In this section, we provide a review of the basic concepts underlying the Terrier Information Retrieval toolkit [Ounis 2007], which we used in our system. For a more detailed explanation, please refer to the Terrier documentation or the Information Retrieval textbook by Manning et al. [2008].

An *inverted index* is the key data structure that enables fast lookup of a document that matches a query of keywords or phrases. The index is created by passing over all the documents in a corpus and collecting crucial bits of information about each individual word. This information is then used to build a *posting list*, which is a table mapping each unique word in the corpus to the set of documents that mention it. To prevent recalculation each time, the total number of documents is stored separately in the index. Additionally, the number of times a word is seen in a document and the number of documents containing a particular word are also stored with each entry. The entries in the index are sorted alphabetically, allowing the index to be stored on disk in a format efficient to search.

Many scoring models are available in Terrier, and we selected the default PL2 scoring function because it has been shown to have relatively good overall early precision[5]. PL2 is a probabilistic model that estimates the importance of a word in a document based on how much its relative frequency diverges from its frequency in the collection as a whole. The intuition behind this model is that high-information words are relatively unique and are only found in isolated (but related) subsets of the entire collection. On the other hand, low-information words will be found throughout the entire corpus following some random distribution.

The exact formula for computing the score is derived from the underlying random distribution chosen. In PL2, a Poisson distribution is used along with several normalization factors useful for this type of task. The first normalization tries to reduce the influence of rare words by including a balancing risk factor that penalizes according to the information gain of the term. In addition to smoothing out the weight of rare words, term frequencies are also renormalized to a standard document length to help provide consistent scores across different queries. The final equation for computing term weights for a document is here.[6]

$$\frac{1}{tfn+1}\left(tfn\cdot\log_2\frac{tfn}{\lambda}+(\lambda-tfn)\right)\cdot\log_2 e+\left(\frac{\log_2(2\pi\cdot tfn)}{2}\right), \qquad (1)$$

where tfn is the normalized term frequency, according to the standard document length (sl) and original document length (dl), defined by the formula $tfn = tf\cdot\log_2\left(1+c\cdot\frac{sl}{dl}\right)$, and tf is the expected frequency of the term given by its total count in the corpus divided by the total number of documents in the collection.

These concepts can then be put together to find and rank a set of documents from a corpus given a new query. Generally, this query is preprocessed[7] to standardize the tokens, such as lowercasing, Unicode normalization, and to remove stop-words that will significantly degrade efficiency with no gain in accuracy. Once the query has been preprocessed, all of the documents matching a remaining query term are returned from the posting list and are scored by the formula:

$$score(d, Q) = \sum_{t\in Q} w(t,d)\cdot qtw, \qquad (2)$$

[5] Early precision is a measure of precision measures that consider the top k documents, where k is small.
[6] See http://terrier.org/docs/v3.0/dfr_description.html for more information.
[7] It is also important to preprocess the documents in a consistent way when constructing the inverted index.

where d is the proposed document, and Q is the set of query terms. $w(t, d)$ represents Eq. (1), and *qtw* is the relative frequency of the term in the query between itself and the most frequent term.

4.2. A Deficient-First Retrieval Model

Using only the information retrieval techniques just described above, it is possible to build a complete end-to-end narrative generation mechanism capable of responding and continuing the user's emerging story in real time. We first create an inverted index of our story corpus where each sentence is treated as a document from the perspective of the information retrieval toolkit. Along with the standard information stored in the posting list that allows efficient scoring, we also store some additional information that will facilitate some of the processing required by our application. This includes the database ID of the sentence, the ID of the corresponding story, and the ID of the next valid sentence. Once the index is created, the first generation algorithm is very simple.

(1) Create a query from the user's story.
(2) Find the *k* most similar sentences in our case library using Terrier.
(3) Return the *next* sentence from each of the corresponding sentences as candidates for the user to choose from.

In step (1), we simply take the user's most recently typed sentence as the query to our database. Following standard IR practices, we use a series of preprocessing steps in order to improve the quality of the search, such as tokenization, stop-word removal, and lowercasing. In steps (2) and (3), a maximum of 10 sentences are returned to the user. In the unlikely event that no similar sentences are found, an error is reported to the user and they are free to try again.

This simple model was first demonstrated by Swanson and Gordon [2008], who showed that this approach could achieve surprisingly good results. This work also showed that using more informative queries (i.e., bigrams) improved the choices presented to the user and the quality of the overall stories. In Section 6, we will reevaluate this model with new data to provide a direct comparison with the other models described in Sections 5.

This section will take a more detailed look at places where the simple IR model breaks down and will investigate what is necessary to do a better job. Figure 5 is a partial story fragment written using *Say Anything*. It is annotated with several link types to illustrate the depth of knowledge required for making correct narrative predictions and to also suggest several simple discourse attributes that are important to the ultimate success of generating coherent responses.

At a basic level, a story is nothing more than a description of the state of the world, how these states change over time, and the effect these changes have on the characters involved. An event calculus that had sufficient breadth to cover all the possible states of the world and the axioms to describe how the world is transformed from one state to another would be an ideal computational representation of a story. The world has infinitely many possible states and axioms, rendering a complete solution virtually impossible. In practice, the number of actual activities, events, and states people typically perform are vastly more limited and the complexity can often be abstracted. Even in this reduced domain, hand-authored formal theories that have sufficient inferential coverage are unlikely to be developed in the near future. The severe lack of breadth in this narrower, but extremely large, domain is one of the primary bottlenecks of most state-of-the-art storytelling systems today. Instead, hope rests on finding a more

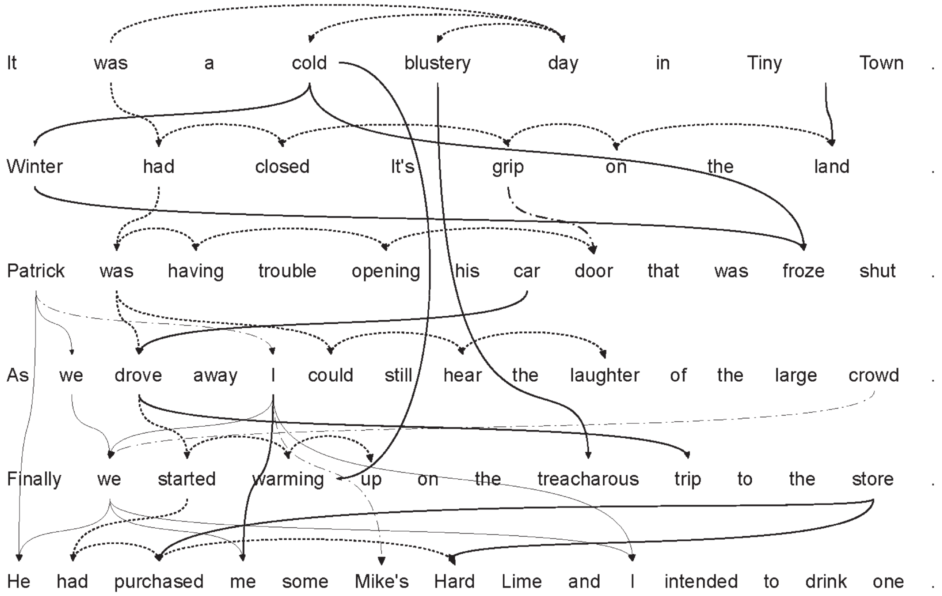

It was a cold blustery day in Tiny Town .
Winter had closed It's grip on the land .
Patrick was having trouble opening his car door that was froze shut .
As we drove away I could still hear the laughter of the large crowd .
Finally we started warming up on the treacharous trip to the store .
He had purchased me some Mike's Hard Lime and I intended to drink one .

Fig. 5. Story graph analysis.

relaxed representation of events that can be automatically learned from raw text, but still has enough of the expressiveness that a more formal theory would provide.

The finely dotted lines in Figure 5 are used to loosely represent partial internal event structure (within a sentence) and the external relationships between consecutive events across sentences. Within a sentence, the links are used to indicate the substructure that encompasses the primary meaning of a specific event (i.e., the dependency structure of the head verb, which could be obtained automatically). For example, the first sentence is an event describing a property of the current state of the world, in particular, that of being a cold and blustery day. In a formal representation this might be expressed as *be(day, cold)* & *be(day, blustery)*. Fortunately, the dependency tree structure used in the figure conveys nearly the same information, but is easier to acquire.

The finely dotted links between sentences represent causal, temporal and other relations between (possibly) separate events. For example, the link between the first and second sentence might specify that the events are contemporaneous, or that one elaborates on the other by providing more detailed information about the same state of affairs. Although long distance dependencies among the events in this story exist (and are not shown in the figure), the local dependencies are enough to show that many types of relationships between events impose a structural order in which these events are narrated. For example, it only makes sense to be warm during the winter if you are somewhere heated, such as in a moving car with the heater turned on. Representing the events of a discourse in the way depicted in Figure 5 suggests learning a theory of event structure could be done empirically and several recent papers have explored different approaches for doing so [Bejan 2008; Chambers and Jurafsky 2008; Manshadi et al. 2008].

With a highly accurate dependency and discourse parser that provides both the sentence-level and document-level relationships, it would be possible to make accurate predictions about changes in the world simply by following the appropriate links in the graph. Although the tools and mechanics are available to extract this information automatically, the relationships are complex, requiring deep understanding or much larger amounts of training data than is available. Unless the utterance was drawn from a common literary trope or widely shared event, it is unlikely the event/state of "purchasing me some Mike's Hard Lime" has ever been described textually before. To maintain tractability, certain concessions must be made to limit the amount of information used, but it comes at the cost of omitting details that are necessary for a correct interpretation. In the first sentence, for example, it is probably not crucial to the state of the world that the cold and blustery day is in "Tiny Town." In the second sentence it is important that "Winter" had closed its grip on the "land," since it might otherwise be confused with gripping a physical object, such as the car door.

Using informative keywords is a simple alternative to using the full dependency structure to make predictions, as illustrated by the dark solid lines in Figure 5. This representation is a generalization of the simple IR-based method described in the previous section. The reason this generation method works so well is because informative words in one sentence are often very indicative of the high-value words in the following sentence. As can be seen in the figure, nearly every sentence has at least one word that is very predictive of another word in the following sentence. For example, "cold" is very predictive of "Winter," "Winter" is very predictive of "froze," "car" is very predictive of "drove" and so on. However, there are also dependencies longer than a single sentence or two, such as "cold" predicting "warming" (or more aptly "warming" being explained by it being "cold"), or the words "treacherous" and "blustery." Even though properly identified keywords tend to be very predictive of each other, they can also lead to some very ill-informed interpretations. For example, a keyword-based system could easily be fooled into relating the "grip" in sentence 2 with the "door" in sentence 3 (dash-dotted line).

The previous paragraphs have emphasized the importance of event prediction in generating a topical continuation of the user's story. There are also other aspects of the discourse that determines the acceptability of a sentence presented to the user. Not only must the candidate sentence be of the appropriate topic, but it must also be structurally correct in terms of its linguistic discourse construction. For example, there can be cases where a sentence is about the right thing, but does not make sense because it fails other discourse constraints. Two of the most common areas where the simple IR algorithm fails are in regards to coreference resolution and in verb agreement. Problems often arise when the main action of a predicted event is exactly what you would expect, but the agency of the characters involved is inconsistent with the previous aspects of the narration. Although the coreference between the entities in Figure 5 can be fairly easily interpreted, the thin solid lines hint at how much world knowledge and experience would be needed to correctly determine what noun phrases refer to the same physical entity. Presumably, "Patrick" and the narrator "I" are distinct individuals, but are both included in the interpretation of "we." It is potentially unclear, without sufficient world knowledge, that the "we" in sentence 5 does not actually refer to the "large crowd" in sentence 4 (dash-dotted line). "He" in the last sentence could be introducing a new character, but is likely to be referring to "Patrick" mentioned three sentences earlier. However, without further processing to identify "Mike's Hard Lime" as a single inanimate object, it would also be difficult to rule out that "He" refers to a person named "Mike."

4.3. An Improved Model Using Discriminative Reranking

The previous section examined some of the crucial elements in accurately predicting the next narrative event in a sequence and also some of the required elements needed to ensure the returned sentence preserves the linguistic integrity of the given narrative. This section examines several modifications of the simple IR-based model that are needed in order to leverage the insights from the previous section. The remainder of this section will describe several new representations of the stories that address some of these issues and the way in which these representations can be used to improve the narrative generation.

Several of these problems are not unique to story understanding and generation. For example, the problem of redundancy in language, where one idea can be expressed synonymously using different words, is a well studied problem. In our case, retrieving sentences that only match particular lexical items will potentially overlook a large number of relevant sentences that mean the same thing but are written with different words. Several methods have been developed in an attempt to address these types of issues. Latent Semantic Indexing [Deerwester et al. 1990], for example, indexes a corpus based on a reduced set of latent topics independent of, but derived from, the lexical words of individual documents. *k*-nearest neighbors is another approach [Smeaton and Van Rijsbergen 1981] that allows documents to be indexed by arbitrary features. Despite the advantages these types of approaches offer, they are not easy to scale to large datasets making the real-time requirements of the system difficult to meet.

What is needed is an approach that has nearly the same retrieval latency as the simple keyword-based approach, but allows a richer set of features to contribute to the final score (and rank) of a candidate sentence. The solution investigated in this section is a two-phase algorithm. The first phase is nearly equivalent to the simple IR-based method described earlier in this section. The output of the first phase is then used as input to a discriminative reranking algorithm that can score the sentences based on an arbitrary set of features, which we think are important to the task. Not only does this model allow for a richer set of discourse and semantic features, but it also allows us to learn from each user interaction and will improve the candidates returned in future turns.

At a high level, the new generation model is similar to the first, with only one additional step.

(1) Generate a query from the user's story.
(2) Find the *n* most similar sentences in our case library using Terrier.
(3) Use the new module to rerank these candidates using discourse and semantic features.
(4) Return the *next* sentence from each of the top *k* corresponding sentences.

Shen and Joshi [2005] survey several ranking and reranking algorithms that have been proposed for problems where the input is a set of candidate elements and the desired result is an ordered list. Many of these algorithms are directly analogous to linear classification algorithms, such as the Perceptron and Support Vector Machines. For example, a maximum margin ranking algorithm finds the set of rankings for the candidates that finds the largest margin between consecutive pairs of candidates, which generally helps performance on unseen data. In contrast, a Perceptron-based ranking algorithm[8] is extremely efficient and can be trained online to update the ranking model after every interaction with the user. Even though max-margin-based approaches usually achieve higher accuracy, we chose a Perceptron based-algorithm because we are interested in the training efficiency and the simple ability to update the model online after every interaction.

[8] Our algorithm is primarily based on the one described in Collins and Duffy [2001], but includes a learning rate and margin.

Similar to the classification approach used in Section 3, the ranking algorithm is a supervised approach that uses a feature vector to represent each instance. In classification, each training instance is labeled with either a +1 or -1, indicating if the example is a positive member of the desired class or not. In ranking tasks, we are given a set of candidates, which are instead labeled with their preferred ordering. The training algorithm uses this ordering to learn a set of parameters that correctly rank the candidates in the training data. These parameters are then used to rank each unseen candidate based on the score computed from its extracted features.

The type of features used to assess the quality of a narrative is critical for the machine learning algorithm to learn a highly discriminative model. In the next section, we discuss the features we investigated for improving the set of candidates suggested by the computer.

4.4. Features for Improving Coherence Through Reranking

One of the core challenges in textual case-based reasoning is mapping the unstructured text to representations that allow as much inferential power as possible for the domain. In this section, we will discuss several intermediate representations that can be constructed from automatic syntactically parsed sentences. These representations try to capture two important aspects for generating coherent sentences. First, we want to try to find and return the most semantically related sentences to the user's story. Second, we want to ensure, regardless of meaning, that the sentence we return is coherent in a structural or grammatical sense.

The most basic representations of our stories simply maps the text to a scalar value. One of these representations is the PL2 retrieval model score (RMS) determined by Terrier. We do not expect a large improvement from this feature, but the supervised nature of the learning algorithm might be able to adjust the weights slightly to improve the performance. Another straightforward representation of our candidates is simply by the number of words they contain (SenLen) and the difference in sentence length between consecutive sentences. Despite the simplicity of this feature we do not expect many large variations in sentence length in a story, which would disrupt the flow of discourse.

One of the key limitations of the simple model introduced in Section 4.2 is the reliance on a single sentence for determining the context and similarity of our textual cases. We would like a representation that takes into account all of the relevant information in the corresponding narratives. Swanson and Gordon [2009a] previously tried incorporating context into their generation model by combining two indexes. The first index modeled each information retrieval document as a sentence. The second index modeled each document as a story prefix, which includes all the sentences up to the sentence used in the first index. Their attempt failed at least in part because they did not have a good way to weight the contribution of each index, leading to many poor selections. In this work, the reranker provides exactly the mechanism we need to find a weight that combines the two models in a more principled and effective manner. To this end, a feature set based on entire story similarity (DocSim) was constructed.

We cannot fully address the problem of linguistic redundancy mentioned earlier, because we are ultimately restricted by the term-based index. However, we did investigate a story representation based on Latent Dirichlet Allocation [Blei et al. 2003], which is a generative language modeling approach that attempts to uncover the latent topics responsible for generating individual words. It is similar to the HMM coherence modeling approach described by Barzilay and Lee [2004]. For our work, we extended the standard LDA model with an additional hidden layer to represent sentence topics. We used Gibbs sampling to estimate the probabilities for each variable in the model, given 100 latent word and sentence topics. The training set consisted of 24,103 documents from the held-out portion of the story corpus. This set of documents consisted of 475,342 sentences, 3,554,440 words, and a vocabulary size of 141,867. The priors were chosen using a linear regression from a large set of trials with parameter variations on a (much smaller) separate set of held out training data.

Three feature sets were extracted from our LDA representation. The first feature set (LDAZ) uses the maximum likely word topic assignments and creates relative frequency unigram and bigram features for the document based on these assignments. Similarly, the exact same process was performed, except the unigrams and bigrams were constructed from the maximum likely sentence topic assignments (LDAΩ). The final feature set (LDAE) extracted the per-word entropy[9] of each document including the candidate sentence using LDA as a probabilistic language model.

So far the representations have primarily focused on the lexical semantics between the user and weblog story. Many of the candidates are not problematic because they are off topic, but due to invalid syntactic or structural constructions, for example, when it is difficult or impossible to interpret the candidate sentence within the context of the user's entire story. Assuming "Patrick" is a boy in Figure 5, it would be difficult to interpret the story if the next sentence after

> *Patrick* was having trouble opening his car door that was froze shut.

was

> As *she* drove away *I* could still hear the laughter of the large crowd.

It is possible to devise a context in which these sentences are acceptable, the shift in pronoun makes this discourse more challenging to interpret. The next several feature sets examine different ways to model the quality of a candidate in terms of its coherence within the user's story alone.

Discourse coherence is a complex relationship between many entities that can often span long distances in a text. Marcu [1997] argues that any globally coherent text must also be locally coherent. Focusing on this easier problem has allowed several tractable computational models of coherence to be developed.

A natural way of capturing this intuition is asking: how likely is the current sentence given the previous sentence? The LDA representation described before is one way of answering this question. Soricut and Marcu [2006] proposed another method using statistical machine translation. In their approach, the words in one sentence are treated as if they were generated by the words in the previous one (or vice versa). They show that this is an effective method for reordering a set of scrambled sentences. As in their work, we use IBM Model 1 [Brown et al. 1993] to derive our features. To train our model we extracted 2,035,966 pairs of consecutive sentences from the same held-out story data described earlier. We used the statistical machine translation toolkit GIZA++ [Och and Ney 2003] to learn the translation probabilities from these sentence pairs.

Two different feature sets were created using the transition probabilities learned from our models. The first set of features (IBM(BG)) was created by extracting all pairs of words across the user's sentence (source) and the hypothesis candidate sentence (target), similar to a bigram. The values for these features were then assigned based on the transition probability of the source and target words. The second feature set (IBM1) was simply the translation probability of the candidate sentence given the previous sentence as calculated by IBM model 1.

[9] Calculated simply as $log_2 p(x)$.

Barzilay and Lapata [2005] introduced a different type of document representation, which they call an *entity grid*, that they show is capable of extracting high-value discourse features from a document. An entity grid (EG) is a matrix representation of a document, where each column represents a unique word in the document and each row corresponds to a sentence. The cells of the matrix contain the dependency relation of the corresponding word (e.g., Subject) in the given sentence. *n*-gram style features are then obtained by traversing down each column and assigning a value to each feature based on its relative frequency in the grid. Each candidate sentence will have different words and dependency relationships assigned to them, causing each entity grid to produce a different set of features and relative frequency distributions.

We also included two additional representations that could be used to estimate the coherence between sentences. The first (COREF) is a structure similar to an entity grid in which there are two columns, one each for the subject and object of a sentence. In this representation, a cell contains the corresponding word matching the column type and sentence. We also employ a similar structure for measuring the distribution of verb tenses across sentences (VA), because we expect most verbs to be past tense and for there to be relatively few shifts in tense between sentences.

4.5. Offline Coherence Evaluation of Discriminative Reranking

In order to train our reranking component we need a set of training instances in which we already know the ordering of each candidate. Fortunately, the design of our system provides an easy way to collect all the training data we need without any extra hand annotation. All we need to do is run the system with the simple information retrieval model and record the interaction history. Every time a user selects a candidate other than the highlighted default candidate (which has the highest score), we have a small training instance in which the order is known. The selected candidate is ranked 1st and the default candidate is ranked second. Given the way English text is read (left-to-right and top-to-bottom), it is assumed that the user has at least read the first sentence. However, due to visual stimuli and other psychological factors, there is no guarantee the user will read the other sentences in any particular order, if they will read them at all.

We gathered our training data by collecting about one thousand stories from dozens of individuals using the simple information retrieval model. During the process of writing these stories people took a total of 5,310 turns writing sentences with the system. Of the 5,310 possible turns, the users selected one of the other 9 alternate sentences 4,395 times. On each of these turns we can assume that one of two things happened. Either the default sentence did not make sense with the story at all (semantically or syntactically) or the user found one of the other sentences more interesting. Our reranking algorithm is agnostic to the particular reason and should work equally well for both.

Following a standard approach, a development, training, and testing dataset were created from the 4,395 pairs of sentences. The development set was composed of 100 stories and a total of 381 sentence pairs. 3305 sentence pairs from 900 stories were included in the training set. The remaining 184 stories and 688 selection candidates were used as test data. Although training the reranker only uses pairs of sentences, during testing the model is applied to all candidate sentences retrieved in the first retrieval phase. Each feature set described in Section 4.5 was applied on its own as well as several other combinations.

The results are summarized in Table I. The table shows the average original rank of the candidate selected by the user (RANK_0) from 0 (best) to 9 (worst). Accompanying

Table I. Average (± stdev) Ratings by the Authors

		Training (CV)			**Testing**		
Features	**#**	**Rank$_0$**	**Rank$_1$**	**%1^{st}**	**Rank$_0$**	**Rank$_1$**	**%1^{st}**
Simple							
RMS	1	4.79	4.21	10.7	4.73	4.27	10.8
SenLen	2	4.79	3.77	15.2	4.73	3.71	16.1
Semantic							
DocSim	3	4.79	3.88	14.5	4.73	3.76	15.3
Coherence							
IBM(BG)	4	4.79	3.82	20.4	4.73	3.96	19.8
IBM1	5	4.79	3.34	16.4	4.73	3.17	15.6
EG	6	4.79	0.36	89.0	4.73	0.33	88.7
COREF	7	4.79	1.50	50.3	4.73	1.40	50.4
VA	8	4.79	3.91	12.1	4.73	3.77	14.8
LDAZ	9	4.79	4.42	10.0	4.73	4.63	8.9
LDAΩ	10	4.79	4.40	11.3	4.73	4.55	10.2
LDAE	11	4.79	3.02	21.8	4.73	2.85	23.8
Combos							
2,3,4,6,7,11	12	4.79	0.40	84.9	4.73	0.36	85.5

this information is the average rank of the selected candidate after it was reranked using the corresponding feature set (RANK$_1$). The percent of selected candidates that were reranked to the top of the collection is also reported (%1st). The results on the left-hand side of the table report results from a 10-fold cross-validation on the training data, with an average of 330.5 tenfold test examples in each fold. The values on the right hand side indicate the performance on the testing data, which contained 688 examples.

It is surprising how well the simple sentence length heuristic works, improving the position of the selected candidate by almost a full rank. Similarly, in light of Swanson and Gordon's [2008] previous attempt, it is reassuring to see that the similarity of the entire document can be used to improve the ranking of a candidate sentence. Entity-grids (EG) are clearly the best performing feature set. It is astounding that they are almost always able to rank the selected candidate sentence at the top of the list. The coreference features are also considerably better than almost every other feature set. In general, the LDA feature sets perform relatively well, however, we did notice a high degree of variability depending on the priors and number of Gibbs sampling iterations used for convergence.

Even though entity grids are the most predictive feature for these offline experiments, it is not 100% clear that using these features in isolation will translate into the most entertaining and usable sentences for the users during live testing (i.e., story writing). For example, the simple IR method may be doing such a poor job that only one reasonable sentence is present among the candidates, and in this case entity-grid features correlate well with this type of sentence. Once the set of candidates is increased beyond 10 in the retrieval phase, more plausible candidate sentences might be found. While entity grids would probably rank them all high, they might lack the fine-grained lexical details that could differentiate them relative to each other. The only true way to find out is to have users write stories with each of the models. This is not the most practical approach in terms of time and resources. For live testing we decided to use feature set 12, because of its excellent performance (0.37 Rank$_1$ on the test set), it combines several disparate types of features, and it is extremely efficient to apply (relative to the other feature combinations).

5. ADAPTATION

Knowledge of human beliefs, activities, and expectations is vital to produce interactive stories that people will find meaningful. Regardless of the size of our corpus we will eventually reach a point of diminishing returns. Linguists often refer to the infinite productivity of language [Lyons 1977], which refers to a human's ability to speak and understand an infinite number of possible utterances. One of the consequences of this productivity is that even two relatively short sentences expressing a similar thought have an incredibly low chance of being communicated using the exact same surface form. This poses a daunting challenge for text-based reasoning systems such as ours.

One way of dealing with this problem is by adapting previously uttered sentences to match the desired meaning of the current discourse, for example, changing the proper name in a sentence whose meaning would be the same otherwise. Adaptation is a fundamental concern of textual case-based reasoning, but it has received the least amount of attention. There has been some recent work in using introspection to adapt the similarity metric [Jayanthi et al. 2010], however, Lamontagne and Lapalme [2004] is the only work we are aware of that has an actual textual revision phase. Their system is an automated email response program for high-volume support centers. Given an email request, they try to identify passages in previous responses that may be relevant to the current inquiry using a text classification algorithm. Certain portions of these responses, primarily named entities, were identified as potential substitution candidates using an information extraction toolkit. A set of rules is then used to decide whether the entity should be replaced by an alternative.

The solution in our framework is to start with a large but finite collection of utterances and define a set of rules that can modify them to fit the user's story more closely. This is a powerful mechanism that could potentially be used to handle deep similarities between texts. For example, consider the following two related stories. First, a story about a young boy who overcomes all odds to start on a major league baseball team by neverending practice and determination. Second, one about a young woman who overcomes many obstacles to become a successful corporate executive by achieving high grades in school and enduring long working hours at her company. At a certain level these stories are significantly different. One focuses on a man's triumphs in sports, while the other is about a woman's corporate success in her professional career. Neither the characters nor their occupations seem to capture the essence of either one of these stories. In both cases, the purpose of these stories is to convey the fundamental proposition that working hard is a critical ingredient to success.

At a surface level these two stories are not particularly good proxies for one another, at least in terms of how they would operate in our narrative generation system. In many cases there would probably be sufficient lexical overlap to rank at least some of the corresponding sentences highly. In many other cases, the entities in the proposed sentences would contain erroneous domain-specific references. For example, both would probably mention phrases like "work hard" and "long hours," but also contain domain-specific words such as "baseball," "games," or "executive." It would be an incredible benefit, drastically reducing the necessary size of the case library, if the relevant aspects of these stories could be extracted and the domain-specific aspects of one could be altered to reflect the domain of the other. For example, one could imagine a mapping between athletic training facilities in the sports domain to an academic institution in the other, or from a baseball stadium to corporate headquarters.

Although this example illustrates the ultimate goal of an effective adaptation system, the types of rules necessary for this translation are complex and not practical at this time. Instead, we focus on adaptation rules that target surface-level aspects of a

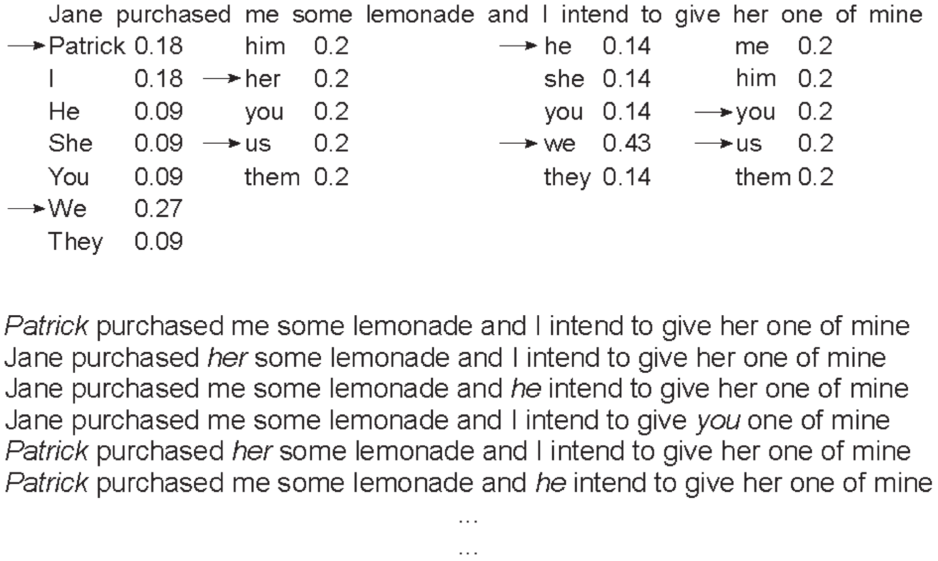

Jane purchased me some lemonade and I intend to give her one of mine
Patrick 0.18
I 0.18
He 0.09
She 0.09
You 0.09
We 0.27
They 0.09
him 0.2
her 0.2
you 0.2
us 0.2
them 0.2
he 0.14
she 0.14
you 0.14
we 0.43
they 0.14
me 0.2
him 0.2
you 0.2
us 0.2
them 0.2
Patrick purchased me some lemonade and I intend to give her one of mine
Jane purchased *her* some lemonade and I intend to give her one of mine
Jane purchased me some lemonade and *he* intend to give her one of mine
Jane purchased me some lemonade and I intend to give *you* one of mine
Patrick purchased *her* some lemonade and I intend to give her one of mine
Patrick purchased me some lemonade and *he* intend to give her one of mine
...
...

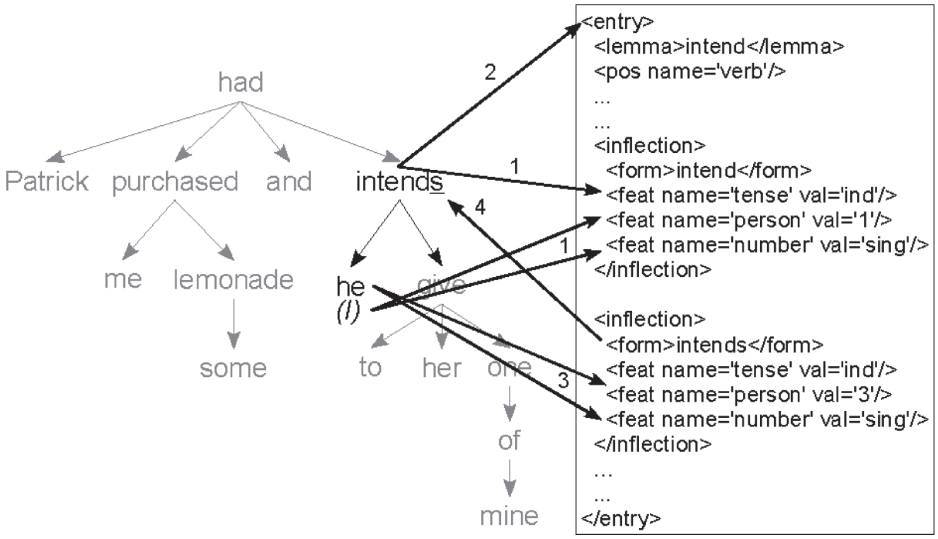
Fig. 6. Step (1) and (2): Potential replacements along with their relative frequency.

sentence, such as pronoun and verb agreement, that we know are problematic but are easier to start with.

The sentence adaptation algorithm is a five step process. Step (1) begins by identifying all of the pronouns and proper names used in the subject or object position in the parse tree. For each identified position a set of valid replacement words is created using a replacement table for each type of word that could appear. The first set of tables correspond to five different classes of pronouns: subjective (e.g., I, he, she, we), objective (e.g., me, him, her, us), reflexive (e.g., myself, himself, herself, ourselves), possessive (e.g., mine, his, hers, ours) and possessive determiners (e.g., my, his, her, our). If the target word is contained within one of these tables, then this set is used for the candidate replacements. Proper names are handled slightly differently. Similar to before, they can also be replaced from a set of pronouns, based on whether the noun functioned as the subject or object. In addition they can also be replaced by other proper names. This is accomplished by maintaining an extra data structure that keeps track of all previous mentions of proper names in the story (i.e., the cast of characters).

Step (2) of the process involves generating a new sentence for a subset of every possible combination of the replacements in each target word set. Unfortunately, the number of combinations for sentences with more than a few target replacement candidates becomes prohibitively large. To prevent the set of candidates from exploding, a simple heuristic was used to limit the total number of possibilities. For any given target word, a maximum of two alternatives were selected as possible replacements. These alternatives were chosen by sampling the entire set of valid possibilities based on each word's relative frequency of occurrence in the entire story. For example, the valid replacements for the pronoun "he" are the subjective pronouns I, we, you, she, one and they. So, if "I" had been seen 4 times, "she" 2 times and "you" 1 time then the relative frequency of each term would be I/0.38, we/0.07, you/0.14, she/0.21, one/0.07 and they/0.07 (add-one smoothing is used to prevent 0s and allow a small chance for any pronoun to be selected). These first steps, (1) and (2), are illustrated in Figure 6.

It is hoped that the characters participating in the events of one of these alternative sentences (or the original) will more closely adhere to the narrative intentions implied



```
<entry>
  <lemma>intend</lemma>
  <pos name='verb'/>
  ...
  ...
  <inflection>
    <form>intend</form>
    <feat name='tense' val='ind'/>
    <feat name='person' val='1'/>
    <feat name='number' val='sing'/>
  </inflection>

  <inflection>
    <form>intends</form>
    <feat name='tense' val='ind'/>
    <feat name='person' val='3'/>
    <feat name='number' val='sing'/>
  </inflection>
  ...
  ...
</entry>
```

Fig. 7. Step 3: Illustration of how verb agreement is fixed using a dictionary.

by the user's story. However, small changes to the subject of a verb can lead to incorrect grammatical agreement between the two. For example, when the subject of the sentence.

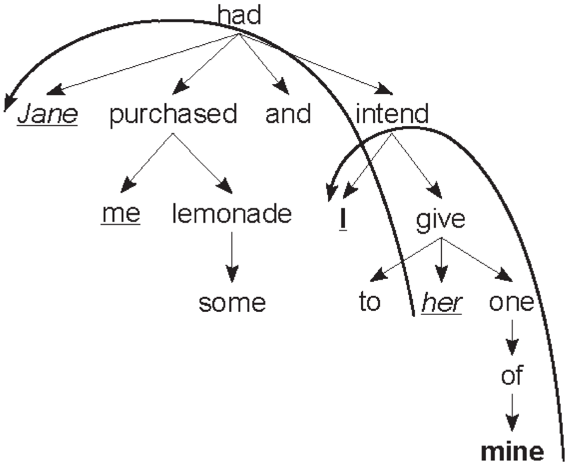
I have more than one lemonade.

is changed to "He," then the new sentence is no longer gramatically correct. Step (3) combats this issue with a special dictionary [Courtois and Silberztein 1989] that provides, where applicable, the number, person, gender, and tense for every lexical entry in the dictionary. The entry containing all lexical variations of any verb whose subject has been adapted is looked up in the dictionary using the number and person information available from the unaltered sentence. The lexical variation corresponding to this entry that matches the number and person information of the new adapted subject is then used to replace the previously ungrammatical verb. Figure 7 gives an example of this process. "I" is the original subject of the verb "intend," but the subject is changed to the pronoun "he," which renders the verb agreement incorrect. A specific entry for "intend" is found in the dictionary by matching the lexical features of the subject (i.e., first person and singular) with the lexical form of the verb as seen in the original sentence. This entry contains many inflected lexical forms depending on the subject's person and number as well as the verb's tense. The new lexical form is determined by using the new person and number (i.e., 3rd person and singular), while maintaining the same tense. The two relevant inflections are shown in the dictionary entry, but the others have been excluded due to space constraints.

Changing the subject and object can also cause more unintended side effects than just incompatible verb agreement. Altering a noun in the sentence can also disrupt the coreference interpretation within the candidate sentence. For example, consider the following sentence:

> Jane$_1$ purchased me$_2$ some lemonade and I$_2$ intended to give her$_1$ one of mine$_2$.

Given only the information present in this sentence, we would expect the coreference between the pronouns to be assigned according to the given indexes. If we change



Patrick purchased me some lemonade and I intend to give **him** one of mine

Jane purchased me some lemonade and ***he*** intend to give her one of **his**

...

Fig. 8. Step (4): Illustration of how the coreference between pronouns is resolved.

“Jane” to “Patrick” to better fit our story from Section 4, then the interpretation changes considerably. Step (4) attempts to preserve the coreference interpretation of the unaltered sentence with the adapted one. To do this we used a coreference resolution algorithm, similar to the one proposed by Hobbs [1976]. Pronouns in the unaltered sentence are resolved in the following way. Starting with pronouns lowest in the parse, the tree is traversed to the closest node in the upper-left portion of the tree that matches the number and gender of the originating pronoun. The gender of proper names was estimated using frequency data collected as part of the 1990 United States census[10]. Using a procedure similar to Step (1), these coreference assignments are used in the adapted sentence to change the pronouns of nodes lower in the tree to correspond with their assigned coreferent higher in the tree if necessary. Once again, this may cause problems with the verb agreement in the sentence and is addressed in the same way as Step (3). A simplified illustration of the process is shown in Figure 8.

The 4 steps outlined provide an adaptation mechanism that addresses some of the most prominent concerns discussed in the previous section. However, there are still a few problems that this approach introduces. First, even with the sampling restrictions in Step (2), the total number of new candidates can prohibit them from being processed in real time. Second, entity grids, the most important feature set during reranking, are not lexicalized. In other words, the changes made to the adapted sentences will have only a small impact on the overall score given by the reranker. This means that the adapted sentences of an unaltered sentence with a high reranking score will also have a high score. In effect, this could end up populating the list of candidates returned to the user with only a single sentence and its minor variations. To avoid these issues, a Step (5) is included for each candidate sentence. In this step all of the adaptation candidates for the sentences are processed by the reranker. The top two of these alternatives, plus the original unaltered sentence, are pushed onto a global set of candidates. After all of the candidates from phase 1 (basic retrieval) have been processed,

[10] http://www.census.gov/genealogy/names/

Table II. Average (+/− stdev) Ratings by the Authors

Model	#Stories	Coherence	Believability	Usability	Entertainment
Unigram	601	3.46 ± 1.11	3.53 ± 1.16	3.08 ± 1.19	3.99 ± 1.05
Bigram	567	3.63 ± 1.11	3.59 ± 1.20	3.27 ± 1.19	4.14 ± 1.02
Reranking	443	3.51 ± 1.15	3.62 ± 1.20	3.85 ± 1.05	4.38 ± 0.83
Adaptation	429	3.46 ± 1.07	3.55 ± 1.19	3.90 ± 1.02	4.33 ± 0.85

the stack of global candidates are reranked, and finally the top 10 are sent to the user as before.

6. EVALUATION

To evaluate the different components of our system we performed a large user study. We used Amazon's Mechanical Turk to crowd-source the authoring of our stories and to collect numerous subjective ratings for each story on several criteria. It has been shown that high-quality annotation and data can be obtain from Mechanical Turk [Callison 2009]. Several steps were taken to help minimize noisy data. First, we placed an 8-sentence (4-turn) minimum requirement on the stories to force some development in the narratives. Second, we required the workers to be located in the United States and to have a minimum task acceptance rate of 97%. Third, every story was manually verified to remove obvious spam, and, finally, several objective verification questions were asked along with our subjective ratings to help identify fraudulent work.

Using this methodology, we were able to collect a total of 2,040 stories and 15,384 independent (approximately 8 per story) ratings. For the simple information retrieval model, we collected 601 stories in which the user's most recent sentence was converted into a unigram query. We also collected 567 stories for this simple model that also included bigram keywords in the query. After collecting these stories, we learned a reranking model as described in Section 4 and collected 443 stories that only used the reranker and 429 stories that used both the reranking and adaptation components.

As mention in Section 2, after completing a story, the authors were asked to subjectively rate their stories and experience on several criteria.

(1) Does the story make sense? (Coherence)
(2) Is the story believable? (Believability)
(3) Did you have fun writing the story? (Entertainment)
(4) How easy was it to write? (Usability)

The independent raters were also asked a similar set of questions except for question 4). The remainder of this section will present the results of the surveys for the authors and independent raters.

6.1. Author Evaluations

Table II presents the results of the authors' subjective ratings on their own stories for the four categories Coherence, Believability, Usability and Entertainment. As in Swanson and Gordon [2009a], using bigram queries provides an overall improvement over unigram queries and is significant ($p < 0.05$) for all categories except Believability. Surprisingly, the authors did not believe their stories were more coherent when using the reranking model, despite its excellent performance in the offline experiments. In fact, there was even a slight decrease from the bigram-only model. Although they rated these stories slightly more believable, the result was not dramatic nor statistically significant. However, there was a large and statistically significant jump in both usability and in entertainment. This result is perplexing. Presumably, if the system is

Table III. Story Authoring Statistics (average value +/− stdev)

Model	Max Len	Avg Len	% Top	MRR	Time(s)	Time(s)/Sen
Unigram	27	9.41 ± 2.31	0.08 ± 0.09	0.36 ± 0.30	460.5 ± 411.8	44.9 ± 32.0
Bigram	25	9.50 ± 2.51	0.09 ± 0.10	0.34 ± 0.29	492.4 ± 463.7	47.9 ± 35.6
Reranking	27	9.54 ± 2.68	0.07 ± 0.08	0.28 ± 0.07	399.2 ± 294.3	40.1 ± 22.8
Adaptation	36	9.63 ± 3.07	0.04 ± 0.04	0.23 ± 0.04	406.1 ± 286.5	39.3 ± 20.6

easier and more fun to use, it is because the responses are more coherent and appropriate. So, some other explanation must be the reason for this discrepancy.

One possibility is that the system is in fact performing better locally. Each sentence returned is much more appropriate given the local context (i.e., the previous sentence only). However, the candidate sentence still fails to be cohesive within the entire story. Another possibility is that there is an inverse (or no) correlation between coherence and entertainment. Sentences could be entertaining because they are incoherent in a way similar to Mad Libs, which stresses the ability to interpret unexpected and often incongruous phrases. In our previous work [Swanson and Gordon 2009b], we found that randomly returning sentences to users produced less entertaining and less coherent stories than the information retrieval models used in that system. Psychological factors may also be another possibility for this dissonance. It seems reasonable to believe that many people enter into their stories with a predefined vision of how it should proceed. So, even though the model may be returning sentences that do make (better) sense for the story that is actually there, they do not conform to the story the author imagines should be there. In general there was no statistically significant difference between the reranking and adaptation models.

In addition to the strictly subjective ratings shown in Table II, several more objective measures were also tracked in Table III. One indirect way to measure how well our system is doing is by counting how many turns the user is taking. Presumably, the number of turns a user takes is a reflection of how engaged they are with the system. We looked at both the maximum number of turns taken with each model as well as the average number of turns. In general, the average number of turns increases as we add functionality to *Say Anything*, but the differences are quite small and not statistically significant. In all, the cases in which the users were writing more sentences (> 9) than they were required to write were written in order to complete the task.

Similar to the offline reranking experiments in Section 4, we looked at how often the user selected the top-ranked candidate (% Top) and the mean reciprocal rank (MRR), which is the average of the reciprocal ranks of the selected candidate. It is surprising that the MRR for the simple information retrieval methods is greater than for the reranking and adaptation models. We believe that the same reasons given for the strange coherence results are also a reasonable explanation for why the ranks are different than expected.

Finally, we also kept track of how long the users were taking to write their stories. The total amount of time in seconds is reported under Time(s) and also the average number of seconds per sentence (Time(s)/Sen) in order to normalize by story length. We initially believed that users would spend more time with the system if it was producing better candidates since they would be able to continue their stories longer. However, as the model complexity increases, the amount of time spent decreases. In combination with the higher usability and entertainment ratings, we now believe the reduced amount of time is more evidence that the quality of the candidates is improving despite the lower coherence ratings. What we suspect is that the user is having an easier time finding a sentence that they like and so it takes them less time in general to complete each turn and, therefore, for the story overall.

Table IV. Average (+/− stdev) Ratings by Independent Readers

Model	Coherence (#Ratings)	Believability (#Ratings)	Entertainment (#Ratings)
Human	3.65 ± 1.24 (0878)	3.70 ± 1.26 (0867)	2.96 ± 1.26 (0867)
Unigram	3.29 ± 1.33 (3901)	3.30 ± 1.32 (3901)	2.84 ± 1.26 (3895)
Bigram	3.42 ± 1.27 (3505)	3.41 ± 1.28 (3509)	2.91 ± 1.23 (3506)
Reranking	3.55 ± 1.24 (3067)	3.51 ± 1.26 (3064)	2.98 ± 1.24 (3058)
Adaptation	3.51 ± 1.22 (3048)	3.48 ± 1.27 (3049)	2.94 ± 1.22 (3042)

6.2. Story Rating Results

The results in the previous section show that certain aspects of the new narrative generation algorithm improve the user's experience with the system as indicated by their subjective ratings and the reduced amount of time needed to write a story without a significant reduction in coherence. The coherence and believability ratings provide a less clear picture of the overall quality of the stories generated with the reranking model. The lack of improvement in these ratings could simply reflect that these stories are not any better (or more coherent) than before. On the other hand, several factors discussed in the previous section could also be at play. This section examines the results of independent ratings by users and will shed some light on which of these scenarios is more likely.

Table IV compares the average ratings from the independent pool of raters for all the models. We see that the reranking and adaptation models are rated significantly higher in coherence and believability than the unigram and bigram models[11] despite having the poorest ratings among the authors. These results seem to offer a different explanation from what was indicated by the authors evaluation of their own compositions. When other readers judge the quality of the narratives written with the reranking models, they are rated significantly higher than the information retrieval methods alone. In fact, the stories written with the reranking models are rated almost as highly on coherence as human-authored stories from our weblog corpus and are even rated more enjoyable to read. These results appear to lend credence to the idea that people are not good at objectively assessing their own stories and providing impartial ratings for coherence judgments. Even though the authors themselves did not rate their own stories higher on coherence, other people clearly thought they were easier and more coherent to read.

The adaptation model was not able to improve over the reranker alone and is even rated slightly below it, although the differences are not statistically significant. The reranking model is successful because it is able to push incoherent sentences out of the list of sentences returned to the user while bringing more coherent sentences to the top. Many of the incoherent sentences that the reranker removes from the list are potentially more appropriate choices for the user except for minor problems with the sentence, such as an incompatible pronoun. The primary motivation of the adaptation component was to try to leverage the best of both worlds. We would like the reranker to find better choices based on a richer set of features than the simple word-based index but still be able to keep the semantically relevant choices by fixing their inconsistencies. In a best-case scenario this should produce a candidate list filled with extremely relevant and coherent sentences. It appears the adaptation component provided the opposite effect. For reasons stated at the end of Section 5, a highly ranked unmodified candidate sentence will probably produce alternative adapted sentences that are also highly ranked. If one sentence makes it into the set returned to the user, the other two variations will also. Even though one of these alternatives is more likely to be coherent with the user's story, it also removes two other distinct options the user can

[11] $p < 0.05$

choose from. It then becomes a question of whether it is better to present the user with a narrow, but more likely correct, selection of relevant candidates or a broader range of possibly incoherent candidates. These results suggest that without a more sophisticated method for adapting the candidate sentences, it is more effective to provide a broader range of sentences using only the reranker.

7. RELATED WORK

TALE-SPIN [Meehan 1976] is generally considered to be the first modern attempt at an automated story generation system, although some template-based systems had come before it. The major insight in this work was a change in representation. Meehan used a formal language to model the story elements, a set of rules for how these elements interacted, and an AI reasoning mechanism to determine how to apply the rules in a given context. To generate a story, one input a set of initial conditions and the system would translate the output of the reasoning engine into simple English sentences. Perhaps without even realizing it, Meehan had separated the story generation process along the lines of the *fabula* and the *sujet*. The prose of TALE-SPIN and other early narrative generation systems could hardly be called artistic by a traditional narrative analysis. Since that time, there has been a substantial improvement in both the prose, for example StoryBook [Callaway and Lester 2002], and the sophistication in which narrative devices are employed to express literary effects, such as suspense in the work of Cheong and Young [2008].

Gervás [2009] surveys several of the most significant advances in automatic story generation beginning with Klein's Novel Writer [1973] and ending with Montfort's Narrator [2007]. These systems use several different mechanisms for generation, including template-based, case-based, and planning. However, all of the systems use hard-coded domain knowledge leading to the authoring and scalability problems discussed earlier in the article.

While interactive narrative shares many of the same properties as the narrative generation systems, significant differences arise because of the temporal and spatial dynamics. Some of the earliest examples of interactive storytelling are the text adventure games Zork [Anderson et al. 1979] and its predecessor Adventure [Crowther and Woods 1977]. There is little to no artificial intelligence to these systems, they, too, model the world using hard-coded domain knowledge using a formal representation. Part of the environment is made available to the user through textual descriptions, such as “You are standing in a forest with a large rock in front of you.” Using textual commands, such as “MOVE ROCK”, one could manipulate the environment in order to progress through the story world and eventually solve the puzzle. Although these systems produce textual descriptions from the underlying domain model, similar to TALE-SPIN, the “story” in these cases is basically a literal transcription of the plot elements uncovered by the user as they discover them. Although most text adventure games still follow this basic model, Montfort's narrator in his system nn [Montfort 2007] is a notable exception that tries to tell the story of the game so far using narrative devices such as point-of-view and flashbacks.

In the overall landscape of digital media, most graphics-based interactive storytelling systems and games are still not much more than pictorial versions of their text-based predecessors. The difficulty in moving beyond the *fabula* in interactive storytelling is a well-know problem in the community and has been coined the “narrative paradox” by Aylett [1999]. That is, once a user is given agency in the process, an apparent conflict between freedom and directed narrative goals arises.

Within the interactive storytelling community there has been a concerted effort to overcome these opposing forces. There are too many to describe in detail, but we will highlight a few representatives that will make it clear where our system falls in the landscape and how our architecture is a departure from the norm. For a more comprehensive review of other interactive storytelling systems, we recommend Arinbjarnar et al. [2009].

Façade [Mateas and Stern 2003] is generally considered one of the first complete media experiences that successfully balanced user freedom, while maintaining a directed and emotionally expressive narrative. In Façade, *beats* are the building blocks of a story. A beat is a short atomic sequence of actions that must be performed together. There is no fixed order in which the beats must occur, but each beat is annotated with a set of preconditions that determine the context in which it is available to the story and a set of effects, which modify the world state when they are executed. A global drama manager monitors the entire state of the experience and tries to pick one of the available beats that will create the best story.

Although Façade was a major step forward, it suffers at least two major drawbacks. First, it presented a natural language interface to the user in order to give the player as much freedom as possible to interact with the environment. Despite this apparent virtue, mapping natural language to a pre-authored domain model is an extremely challenging task, and this often led to system misinterpretations. These errors in turn usually caused the players to feel less in control. Second, it took three man-years to author Façade, which produced a 20–25 minute playable experience that has enough variability to be replayed about 3–5 times. Although this amount of effort may be within the same scope of other large media productions, such as films, it highlights the difficulty required to create such an experience. Furthermore, it is unclear how this type of approach scales to larger domains. Would a 40–50 minute experience take 6 man-years to author? What about one with twice the number of non-player characters? A case could be made that economies of scale, reusable components, and familiarity with the system could lead to a decrease in authoring time. However, practical experience in similar authoring environments seems to indicate that a larger web of preconditions and effects dramatically increases the complexity of authoring coherent sequences and the debugging of unexpected outcomes.

Many other architectures for interactive storytelling have also been proposed, that differ significantly in the way they model user interaction and story generation. Classical planning-based approaches, such as Mimesis [Riedl et al. 2008], have become popular because of their ability to model complex causal and temporal structures. Search-based drama management is an alternative method for real-time narrative planning that heuristically explores and evaluates a bounded depth in the search space. This method was first proposed by Weyhrauch [1997] and has been further explored by Cavazza et al. [2002] and Nelson and Mateas [2005]. Probabilistic and decision-theoretic approaches [Mott and Lester 2006] are also alluring because of their ability to model uncertainty and incomplete information. There are also several case-based interactive storytelling systems that have been developed, most notably MEXICA [Pérez Y Pérez 1999], OPIATE [Fairclough 2004], work by Gervás et al. [2005], Sharma et al. [2007], and Hajarnis et al. [2011].

Although these approaches have considerable differences, there is one common theme among nearly all of them. In each case, a domain model must be authored by hand in the formal representation of choice. This is done with beats in Façade, STRIPS-like operators in planning approaches, Bayesian networks (or other graph structures) in probabilistic approaches, and other hybrid formats used in systems, such as MEXICA. Creating new story environments in these systems requires an expert in the particular knowledge representation format and the same pitfalls discussed in relation to Façade are equally concerning. Requiring expert domain engineers also poses another problem. It can be unclear if the success of a system is due to a superior system architecture or if it is due to the expertise of the knowledge engineer in formulating a well designed domain.

8. CONCLUSIONS

Interactive narrative is a genre of storytelling that allows active participation by the reader to affect the development and outcome of the story. It is a highly compelling medium because of the fundamental connection between stories and human nature. However, this also makes it extraordinarily challenging because of the breadth and depth of common sense knowledge required to understand and produce compelling event sequences when a participant is allowed to change the narrative world around them.

Section 7 discussed several typical ways that researchers have attempted to tackle these problems. These approaches usually model the major events (i.e., scenes or beats) of the narrative in a formal language, such as planning operators. Stories are generally produced from these representations using some kind of planning mechanism, for example, a partial-order causal-link planner [Riedl and Young 2010] or heuristic search in task networks [Cavazza et al. 2002]. As the user changes the state of the world through his or her interaction, the planning algorithm can reformulate the goals of the story and adjust its direction to suit the events as they have actually unfolded.

This methodology is reasonable, intuitive, and it has been shown to be effective for many limited domain use cases. However, a major component of its operation and success remains a persistent roadblock for common sense reasoning about real-world stories. Where does the content come from and how do we know the causal and temporal structure of the scenes? In traditional systems this type of knowledge acquisition is done explicitly by the authors. However, the breadth and depth of common sense knowledge required to reason about stories is composed of vast amounts of seemingly obvious and boring pieces of information about people's everyday lives and experiences. We as humans acquire so much of this knowledge subconsciously through our own experiences that we do not even realize that we know it or that it is even important. Even with breakthroughs in intelligent user interfaces to aid in the authoring of interactive stories [Cheong et al. 2008; Mehta et al. 2009; Pizzi and Cavazza 2008], it remains an open question whether the necessary knowledge can be acquired by hand.

Standard interactive storytelling systems have primarily focused on detailed models of narrative tropes, but there is not a clear vision of how these approaches can scale to worlds that have the richness of human experience. Our work offers an alternative approach for the large-scale knowledge acquisition required to give the user a high degree of freedom and depth in an interactive experience. Instead of relying on a small number of domain-specific axioms, we leverage the experiences of millions of people to solve this authoring problem. From a technical perspective, our approach is a departure from most other interactive storytelling system. Despite the seemingly stark differences, the ultimate goal of the systems is the same. We are striving to provide a virtual story world that allows the user to manipulate the environment, while maintaining a coherent narrative structure for the entire experience. In the future we hope the benefits of goal-oriented long-term planning can be merged with our large-scale data-driven approach to achieve more focused and structured experiences.

REFERENCES

ANDERSON, T., BLANK, M., DANIELS, B., AND LEBLING, D. 1979. Zork. In *Proceedings of Infocom*.

ARCOS, J. L., DE MANTARAS, R. L., AND SERRA, X. 1998. Saxex: A case-based reasoning system for generating expressive musical performances. *J. New Music Res. 27* 3, 194–210.

ARINBJARNAR, M., BARBER, H., AND KUDENKO, D. 2009. A critical review of interactive drama systems. In *Proceedings of the Symposium on Artificial Intelligence and Simulation Behavior: AI and Games*.

AYLETT, R. 1999. Narrative in virtual environments-towards emergent narrative. In *Working Notes of the Narrative Intelligence Symposium*.

BARZILAY, R. AND LAPATA, M. 2005. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Ann-Arbor, 141–148.

BARZILAY, R. AND LEE, L. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In Human Language Technologies - North American Association of Computational Linguistics. Boston, MA. In *Proceedings of the Conference of the Association for Computational Linguistics - Human Language Technologies*.

BEJAN, C. A. 2008. Unsupervised discovery of event scenarios from texts. In *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*.

BLEI, D. M., NG, A. Y., AND JORDAN, M. I. 2003. Latent Dirichlet allocation. *J. Machine Learn. Res. 3*, 993–1022.

BLOOD, R. 2000. Weblogs: A history and perspective.
http://www.rebeccablood.net/essays/weblog_history.html.

BROWN, P. F., PIETRA, V. J. D., PIETRA, S. A. D., AND MERCER, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computat. Linguist. 19*, 2, 263–311.

BURKE, R. D., HAMMOND, K. J., KULYUKIN, V. A., LYTINEN, S. L., TOMURO, N., AND SCHOENBERG, S. 1997. Question answering from frequently asked question files: Experiences with the FAQ finder system. Tech. rep., University of Chicago.

CALLAWAY, C. B. AND LESTER, J. C. 2002. Narrative prose generation. *Art. Intell. 139*, 2, 213–252.

CALLISON-BURCH, C. 2009. Fast, cheap, and creative: Evaluating translation quality using Amazon's Mechanical Turk. In *Proceedings of the Conference on Empirical Methods in Natural Language*. Association for Computational Linguistics, 286–295.

CARLSON, L., MARCU, D., AND OKUROWSKI, M. E. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*. 1–10.

CAVAZZA, M., CHARLES, F., AND MEAD, S. J. 2002. Interacting with virtual characters in interactive storytelling. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*. 318–325.

CHAMBERS, N. AND JURAFSKY, D. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Association for Computational Linguistics - Human Language Technologies*.

CHEONG, Y.-G. AND YOUNG, R. M. 2008. Narrative generation for suspense: Modeling and evaluation. In *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling*. 144–155.

CHEONG, Y.-G., KIM, Y.-J., MIN, W.-H., SHIM, E.-S., AND KIM, J.-Y. 2008. PRISM: A framework for authoring interactive narratives. In *Interactive Storytelling*, 297–308.

COLLINS, M. AND DUFFY, N. 2001. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. 263–270.

COURTOIS, B. AND SILBERZTEIN, M. D. 1989. *Dictionnaires électroniques du français*. Larousse, Paris.

CROWTHER, W. AND WOODS, D. 1977. Adventure.

CUNNINGHAM, P. 2009. A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Trans. Knowl. Data Engin. 21*, 11, 1532–1543.

DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. 1990. Indexing by latent semantic analysis. *J. Amer. Soc. Info. Sci. 41*, 391–407.

DREDZE, M., CRAMMER, K., AND PEREIRA, F. 2008. Confidence-weighted linear classification. In *Proceedings of the 25th International Conference on Machine Learning*. 264–271.

FAIRCLOUGH, C. 2004. Story games and the OPIATE system. Ph.D. thesis, University of Dublin - Trinity College.

GERVÁS, P. 2009. Computational approaches to storytelling and creativity. *AI Mag. 30*, 3, 49–62.

GERVÁS, P., DÍAZ-AGUDO, B., PEINADO, F., AND HERVÁS, R. 2005. Story plot generation based on CBR. *J. Knowl. Based Syst. 18*.

GILLICK, D. 2009. Sentence boundary detection and the problem with the U.S. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, *Human Language Technologies*. (Companion Volume: Short Papers). 241–244.

GORDON, A. S. AND GANESAN, K. 2005. Automated story capture from conversational speech. In *Knowledge Capture*, 145–152.

GORDON, A. S. AND HOBBS, J. R. 2004. Formalizations of commonsense psychology. *AI Mag. 25*, 4.

GUYON, I. AND ELISSEEFF, A. 2003. An introduction to variable and feature selection. *J. Machine Learn. Res. 3*, 1157–1182.

HAJARNIS, S., LEBER, C., AI, H., RIEDL M. O., AND RAM, A. 2011. A case base planning approach for dialogue generation in digital movie design. In *Proceedings of the International Conference on Case-Based Reasoning*.

HAMMOND, K. J. 1989. *Case-Based Planning: Viewing Planning as a Memory Task*. Academic Press Professional, Inc.

HAYES, P. J. 1985. Naive physics I: Ontology for liquids. In *Formal Theories of the Commonsense World,* J. R. Hobbs and R. C. Moore Eds. Ablex Pub, 71–108.

HOBBS, J. 1976. Pronoun resolution. Tech. rep., Department of Computer Sciences, City College, City University of New York.

HOBBS, J. AND SAGAE, A. 2011. A commonsense theory of microsociology: Interpersonal relationships. logical formalizations of commonsense reasoning. In *Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium*.

JAYANTHI, K., CHAKRABORTI, S., AND MASSIE, S. 2010. Introspective knowledge revision in textual case-based reasoning. In *Case-Based Reasoning. Research and Development*. I. Bichindaritz and S. Montani Eds., Springer, Berlin, 171–185.

KLEIN, S., AESCHLIMAN, J. F., BALSIGER, D., CONVERSE, S. L., COURT, C., FOSTER, M., LAO, R., OAKLEY, J. D., AND SMITH, J. 1973. Automatic novel writing: A status report. Tech. rep. 186, Computer Science Department, The University of Wisconsin, Madison, WI.

LAMONTAGNE, L. AND LAPALME, G. 2004. Textual reuse for email response. In *Advances in Case-Based Reasoning*, P. Funk and P. A. González Calero Eds., Springer, Berlin, 242–256.

LYONS, J. 1977. *Semantics*: Vol. 1 Rev. Cambridge University Press.

MAJEWSKI, J. 2003. *Theorising Video Game Narrative. Master of Film and Television*. Bond University, Australia.

MANGIONE, J. 1996. *The Dream and the Deal: The Federal Writers' Project, 1935–1943*. Syracuse University Press.

MANNING, C. D., RAGHAVAN, P., AND SCHÜTZE, H. 2008. *Introduction to Information Retrieval* 1st Ed. Cambridge University Press.

MANSHADI, M., SWANSON, R., AND GORDON, A. S. 2008. Learning a probabilistic model of event sequences from internet weblog stories. In *Proceedings of the 21st International Conference of the Florida AI Society, Applied Natural Language.* (Processing Track).

MARCU, D. 1997. The rhetorical parsing of natural language texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics*. 96–103.

MATEAS, M. AND STERN, A. 2003. Façade: An experiment in building a fully-realized interactive drama. In *Proceedings of the Game Developers Conference*.

MEEHAN, J. R. 1976. The metanovel: Writing stories by computer. Ph.D. thesis, Yale University.

MEHTA, M., ONTAÑÓN, S., AMUNDSEN, T., AND RAM, A. 2009. Authoring behaviors for games using learning from demonstration. In *Proceedings of the International Conference on Case-Based Reasoning*. Workshop on Case-Based Reasoning for Computer Games.

MILAM, D., EL-NASR, M. S., AND WAKKARY, R. 2009. A Study of interactive narrative from user's perspective. In *Handbook of Digital Media in Entertainment and Arts*, B. Furht Ed., 653–681.

MILTSAKAKI, E., PRASAD, R., JOSHI, A., AND WEBBER, B. 2004. The penn discourse treebank. In *Language Resources and Evaluation*.

MONTFORT, N. 2007. Generating narrative variation in interactive fiction. Ph.D. thesis, University of Pennsylvania.

MOTT, B. W. AND LESTER, J. C. 2006. U-director: A decision-theoretic narrative planning architecture for storytelling environments. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 977–984.

MOTT, B., LESTER, J., AND BRANTING, K. 2005. The role of syntactic analysis in textual case retrieval. In *Proceedings of the Textual Case-Based Reasoning Workshop*. R. Weber and L. K. Branting Eds., 120–127.

NELSON, M. AND MATEAS, M. 2005. Search-based drama management in the interactive fiction anchorhead. In *Proceedings of the 1st Artificial Intelligence and Interactive Digital Entertainment Conference.*

OCH, F. J. AND NEY, H. 2003. A systematic comparison of various statistical alignment models. *Computat. Linguist. 29*, 1, 9–51.

OUNIS, I. 2007. Research directions in Terrier: A search engine for advanced retrieval on the Web. *Novatical UPGRADE* (Special Issue on Web Information Access).

PÉREZ Y PÉREZ, R. 1999. MEXICA: A computer model of creativity in writing. Ph.D. of dissertation. In *Proceedings of the AISB Symposium on Creative Language.* 46–51.

PIZZI, D. AND CAVAZZA, M. 2008. From debugging to authoring: Adapting productivity tools to narrative content description. In *Proceedings of the 1st Joint International Conference on Interactive Digital Storytelling*. Springer-Verlag, 285–296.

PUSTEJOVSKY, J., HANKS, P., SAURÍ, R., SEE, A. GAIZAUSKAS, R., SETZER, A., RADEV, D., SUNDHEIM, B., DAY, D. FERRO, L., AND LAZO, M. 2003. The TIMEBANK corpus. In *Corpus Linguistics*, Lancaster University, Lancaster UK.

RECIO-GARCÍA, J. A., DÍAZ-AGUDO, B., AND GONZÁLEZ-CALERO, P. A. 2007. Textual CBR in jCOLIBRI: From retrieval to reuse. In *Proceedings of the International Conference on Case-Based Reasoning Workshop on Textual Case-Based Reasoning*.

REYNAR, J. C. AND RATNAPARKHI, A. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing*.

RIEDL, M. O. AND YOUNG, R. M. 2010. Narrative planning: Balancing plot and character. *J. Art. Intell. Res. 39*, 1, 217–268.

RIEDL, M. O., STERN, A., DINI, D. M., AND ALDERMAN, J. M. 2008. Dynamic experience management in virtual worlds for entertainment, education, and training. *Int. Trans. Syst. Scie. Appl. 3*, 1, 23–42.

RIESBECK, C. K. AND SCHANK, R. C. 1989. *Inside Case-Based Reasoning* 1st Ed. Lawrence Erlbaum.

SAGAE, K. AND LAVIE, A. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics* (Main Conference Poster Sessions). 691–698.

SHARMA, M., MEHTA, M., ONTAÑÓN, S., AND RAM, A. 2007. Player modeling evaluation for interactive fiction. In *Proceedings of the 3rd Artificial Intelligence for Interactive Digital Entertainment Conference Workshop on Optimizing Player Satisfaction*.

SHEN, L. AND JOSHI, A. K. 2005. Ranking and reranking with perceptron. *Machine Learn. 60*, 1–3, 73–96.

SMEATON, A. F. AND VAN RIJSBERGEN, C. J. 1981. The nearest neighbour problem in information retrieval: An algorithm using upperbounds. In *Proceedings of the 4th Annual International ACM SIGIR Conference on Information Storage and Retrieval: Theoretical Issues in Information Retrieval*.

SORICUT, R. AND MARCU, D. 2006. Discourse generation using utility-trained coherence models. In *Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics* (Main Conference Poster Sessions). 803–810.

STEINITZ, R. 1997. Writing Diaries, Reading Diaries: The Mechanics of Memory. *The Communication Review*, 2, 43–58.

SWANSON, R. AND GORDON, A. 2008. Say Anything: A massively collaborative open domain story writing companion. In *Proceedings of the 1st International Conference on Interactive Digital Storytelling*.

SWANSON, R. AND GORDON, A. S. 2009a. A comparison of retrieval models for open domain story generation. In *Proceedings of the Association for the Advancement of Artificial Intelligence Spring Symposium on Intelligent Narrative Technologies II*.

SWANSON, R. AND GORDON, A. S. 2009b. Say Anything: A demonstration of open domain interactive digital storytelling. In *Proceedings of the International Conference on Interactive Digital Storytelling*.

TECHNORATI.COM. 2008. State of the blogosphere 2008. http://technorati.com/blogging/state-of-the-blogosphere/.

WEBER, R. O., ASHLEY, K. D., AND BRÜNINGHAUS, S. 2006. Textual case-based reasoning. *Knowl. Engin. Rev. 20*.

WEYHRAUCH, P. W. 1997. Guiding interactive drama. Ph.D. thesis, Carnegie Mellon University.

WIRATUNGA, N., LOTHIAN, R., CHAKRABORTI, S., AND KOYCHEV, I. 2005. Textual feature construction from keywords. In *Proceedings of the International Conference on Case-Based Reasoning Workshops*. 110–119.

Received May 2011; revised December 2011, April 2012; accepted June 2012