Data-Driven Alibi Story Telling for Social Believability

Boyang Li, Mohini Thakkar, Yijie Wang, and Mark O. Riedl
School of Interactive Computing
Georgia Institute of Technology
{boyangli, mthakkar, yijiewang, riedl}@gatech.edu

ABSTRACT
As computer games adopt larger, more life-like virtual worlds, socially believable characters become progressively more important. Socially believable non-player characters (NPCs) must be able to act in social situations and communicate with human players. In this paper, we address one aspect of social believability: the construction and telling of alibi stories, or an artificial background that explains what a character has been doing while not in the presence of the human player. We describe a technique for generating alibi stories and communicating the alibi stories via natural language. Our approach uses machine learning to overcome knowledge engineering bottlenecks necessary to instill intelligent characters with social behavioral knowledge. Alibi stories are subsequently generated from learned social behavioral knowledge. By leveraging the Google N-Gram Corpus and Project Gutenberg books, natural language is generated with a discourse planner and text generation that incorporate different expressivity and sentiment, which can be employed to create NPCs with a variety of personal traits.

Categories and Subject Descriptors
I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Games*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language generation*

General Terms
Algorithms, Design, Human Factors.

Keywords
socially believable characters, alibi generation, natural language generation.

1. INTRODUCTION
Many computer games invite players to temporarily suspend their disbelief and enter a rich fictional world populated with non-player characters (NPCs). To create the illusion that these NPCs lead their own lives in the virtual world and are not just part of a show that disappears when the player looks away, each NPC needs to have a background story or an *alibi*, which can explain where they have been and what they have done while they are not with the player [24]. NPCs should be able to tell these background stories and recall details to substantiate their stories when asked to. Otherwise, the suspension of disbelief may quickly fade. Social believability is thus partially a problem of automated story generation.

The benefits of NPCs with background stories are not limited to only computer games. For example, Bickmore and Schulman [4] found that the ability to tell autobiographical stories increases the likelihood that human users will interact with virtual characters over an extended period. This can be advantageous when we need to encourage users to keep interacting with some programs or electronic devices, such as educational software or medical devices for self-monitoring.

In this paper, we tackle two challenges of supporting social believability in NPCs: generating socially believable behaviors and communicating alibi stories according to personal traits. To acquire social believability, behaviors in the alibi stories must adhere to socio-cultural norms in the virtual world. For example, in modern American restaurants, drinks are typically ordered before food. Further, NPCs with different personal traits may tell the stories with different language and levels of detail. Typically, knowledge about how social situations unfold is encoded as a set of scripts—knowledge structures that explain what to do in certain situations and when to do it. However, manually authoring sufficient script knowledge for NPCs with different personal traits is an expensive process. This authoring bottleneck limits the amount of domain knowledge and personal variation available to human-agent interaction and threatens suspension of disbelief.

We create socially believable NPCs by incorporating data about the real world into the game world and character decision-making. These data provide an intelligent agent with observations about how the real world works and the language used by real humans, from which believable behaviors can be derived. Our previous work [11, 12] describes an approach to story generation that learns about *a priori* unknown social situations from exemplar stories acquired using crowdsourcing. This paper applies the technique to alibi generation and tackles the problem of communicating alibi stories via natural language. Making use of the Google N-Gram corpus [16] and books from Project Gutenburg (http://www.gutenberg.org), we offer methods to tune the alibi stories with different levels of detail, and to tell the stories with different styles and sentiments. The combination of these tunable parameters allows NPCs to speak with differing personal traits when telling their alibi stories of the same situation.

2. BACKGROUND AND RELATED WORK

2.1 Social Non-Player Characters
Humans respond to virtual agents much the same way they respond to each other [7, 22]. There are numerous attempts to create intelligent virtual agents that are capable of interacting with humans in social contexts, including entertainment [2, 5], learning [27], and healthcare [3]. There are many aspects to social believability that must be addressed, including theory of mind [25], emotion [6], and storytelling [4].

In computer games, there has been a push toward more socially believable NPCs. A believable NPC should lead an independent life in the virtual world, but simulating every NPC and their interactions is computationally very expensive. To reduce the computational cost, alibi generation [23, 24] is proposed to create a believable life for an NPC on demand. Sunshine-Hill and Badler [24] generate behaviors for pedestrian NPCs when they are being observed. These behaviors are statistically very similar to a full simulation, so players will not realize they are not completely simulated. Robertson and Young [23] propose a story planner that can replan historical events that the player is not aware of, so that alibis remain consistent with the player's knowledge. Our approach to alibi generation differs from previous approaches in that we produce socially believable alibis by using socio-cultural norms encoded in plot graphs, and focus on re-telling of alibi stories with different personal traits. Aligning the generated stories with player knowledge is left for future work.

Other recent work in game artificial intelligence has focused on social interactions with virtual characters. *Prom Week* [15] is an excellent example of a social simulation game in which the player must navigate a character through the social situations surrounding a high school prom dance. *Prom Week* requires over 5,000 rules to capture the associated social dynamics. *The Restaurant Game* [20] attempts to overcome knowledge authoring bottlenecks by learning the social conventions of going to a restaurant from a large number of traces of human behavior in a simulated restaurant environment. Although the technique has been demonstrated to learn a procedure for going to a restaurant, it requires a simulation environment to be built in advance, limiting learning to the situations that have been pre-specified. *SayAnything* [26] overcomes the authoring bottleneck by generating stories from snippets of natural language mined from Web Blogs. However, it requires human intervention to maintain story coherence.

2.2 Natural Language Generation

Natural language generation is the process of planning discourse and then instantiating the discourse in natural language. In the context of computer games, the pragmatic decisions of discourse structure and word choice is important in distinguishing characters from each other by the way they speak. The authoring of a large set of distinguishable characters is generally intractable, but the topic of automatically generating distinct linguistic pragmatics is not well explored. One exception is the work by Lin and Walker [13] on mining linguistic personality traits from corpora of dialogue.

We use a bag-of-word model to determine the emotion in sentences generated as candidates for a character to speak. There are several ways to build a dictionary for emotional values of words: expert annotation, crowdsoucing, and automation. Expert annotations are usually highly precise but cover only a small number of words. Crowdsourcing covers more words with possibly noisy inputs from amateur labelers (e.g., [18]). The automatic approach starts with a few seed words with known values and expands them through relationships between words. It covers a large number of words but sacrifices accuracy. A common drawback of sentiment analysis is lack of sensitivity to word sense based on context. The SentiWordNet lexicon [1] expands on WordNet [17] by annotating synsets (word senses) with values indicating the extent to which each word sense is positive, negative, or objective. SentiWordNet is automatically constructed and not always accurate. We correct and extend SentiWordNet by propagating sentimental values to neighboring

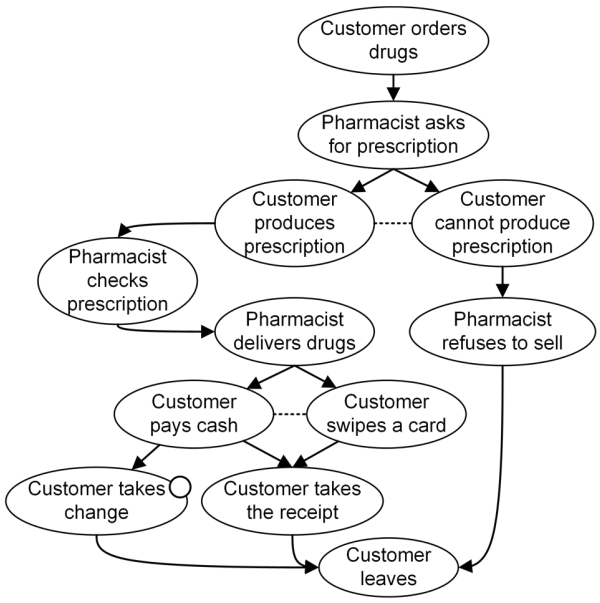

Figure 1. An example plot graph of the pharmacy situation. Vertices are events. Direct edges denote temporal orderings. Dashed lines denote mutual exclusion relations, and a small circle denotes optional events.

words in a corpus of literature texts. Our approach is similar to that by Mohammad [19] and Perrie *et al.* [21], but we use full texts from Project Gutenberg instead of 5-grams from Google N-Gram to incorporate more context. Lu *et al.* [14] proposed a supervised approach that can find words carrying domain-specific sentiments (e.g. "private" is positive for hotel reviews). We use fictional texts to find sentiments in the domain of fiction.

2.3 Learning from Exemplar Stories

The work in this paper builds off the Scheherazade system [11, 12], which learns the structure of events in a given situation from crowdsourced exemplar stories describing that situation. As opposed to other story understanding and story generation systems, Scheherazade is a just-in-time learner; if the system does not know the structure of a situation when it is called for, it attempts to learn what it needs to know from a crowd of people on the Web. This results in a script-like knowledge structure, which we refer to as a *plot graph*. The graph contains events that can be expected to occur, temporal ordering relations between events, and mutual exclusions between events that create branching alternatives. Figure 1 shows an example plot graph for the pharmacy situation, which defines a space of possible pharmacy interactions between a patron and a pharmacist. Possible ways that the encounter can unfold include producing a prescription or not, and paying with cash or credit card. If the patron pays cash, taking back change is an optional event. Each plot graph can be thought of as a compact model of all possible total orderings of events that are believed to be able to occur in the context of a common activity, suitable for modeling social situations involving interpersonal interactions and turn taking.

The exemplar stories from which the plot graph is learned are crowdsourced from Amazon Mechanical Turk. Each crowd worker is asked to write a story describing a given situation with given character names. In order to simplify natural language processing, they are asked to use simple language. For example,

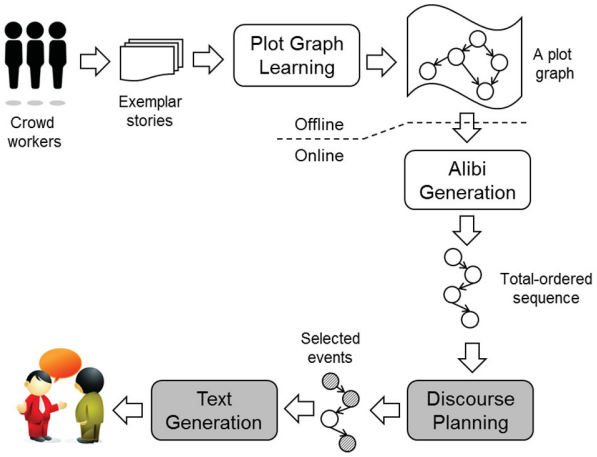

Figure 2. The alibi story telling process.

each sentence should describe a single event with a single verb. No complex or compound sentences and no pronouns are allowed. Each story is compensated for $0.6 to $1. A plot graph can usually be learned from 60 to 80 such stories. Crowdsourcing provides a low-cost and intuitive method for authoring knowledge needed for conversational agents; storytelling can be used to convey tacit knowledge difficult to articulate even for experts [9]. The story-writing task does not require any training in computer science, which is in contrast to, for instance, letting workers write production rules or manipulate graphical models. Hence, turning knowledge engineering into story writing simplifies the task and helps to increase the number of potential participants and lower the cost of hiring.

The learning of the plot graph proceeds in four steps. The first step puts sentences having similar semantic meaning into the same cluster. These clusters become events. The second and third steps identify the temporal orderings between events and mutual exclusions respectively. Finally, we identify the optional events. Interested readers are referred to previous publications [11, 12] for system and algorithmic details. In this paper, we select sentences in the event clusters to describe the events and use the graph structure to determine the importance of each event.

Story generation in Scheherazade is the process of selecting a linear sequence of events from the set of all possible event sequences described by a plot graph. Scheherazade performs story generation by selecting a set of events that do not violate any temporal or mutual exclusion relations in the script [12]. An alibi story is a single, complete event sequence that is presumed to have happened in the virtual world. However, believably *telling* an alibi story requires the consideration of two additional challenges addressed for the first time in this paper: discourse planning—the selection of a subset of events from an alibi story—and natural language generation.

3. SOCIALLY BELIEVABLE DISCOURSE

This section describes the process of generating alibi stories and telling alibi stories in natural language with a variety of personal traits. Our architecture for alibi generation and communication is shown in Figure 2. Plot graph learning is typically an offline process that incrementally constructs a knowledge base of models of social situations that an agent knows how to generate stories about. Plot graph learning is described in detail in [11, 12]. An alibi can be generated on any topic that can be expressed by human crowd workers. For example, an NPC could construct an alibi about going on a date with her virtual boyfriend, going to a restaurant, or witnessing a bank robbery. If a plot graph model doesn't exist for the alibi topic, the plot graph learner will be invoked at the time that the alibi is generated. Given the existence of a plot graph on the topic of the alibi, an alibi is generated as one possibly totally ordered sequences of events that comprise an artificial memory of an NPC's experience. The process of generating the totally ordered sequence is described in detail in [12]. This paper focuses on the last two stages of the architecture: discourse planning and text generation (shown as shaded boxes), which are explained in the following sections.

3.1 Discourse Planning

When humans describe their experiences, they usually do not include every event; some events are too obvious or too mundane to tell. For instance, the event of sitting down in the auditorium is assumed to have happened when someone tells you she watched a movie in a movie theater. However, Scheherazade learns scripts that include most events in the situation and thus generates overly verbose stories [12]. Believable social agents must be able to differentiate unimportant events from important events in the situation, and selectively tell her experience.

The importance of events allows us to perform discourse planning. We can produce a high-level summary of an alibi story by selecting the k most important and k least important events to produce a story of $2k$ length. The k most important events provide *landmarks*—events that help the hearer to understand to what point in the script the storyteller has progressed. The k least important events are those that are most rare and therefore most surprising [11]. The following describes our algorithm, *EventRank*, which determines the importance of events in an alibi generated from a plot graph.

EventRank considers the size of the event cluster, the structure of temporal orderings, as well as mutual exclusion relations, to determine the importance of each event to the telling of the alibi. The algorithm is inspired by the Personalized PageRank algorithm [8], which computes the importance of n vertices contained in a strongly connected directed graph structure, captured as an $n \times n$ transition matrix A, where an entry A_{ji} denotes the probability of transiting from node i to node j. For a vertex i with out-degree d_i, if there is a directed link from node i to node j, we let the corresponding matrix entry $A_{ji} = 1/d_i$. Otherwise, $A_{ji} = 0$. That is, we can transit away from each vertex with equal probability along each of its outgoing edges. It can be shown that the matrix A has an eigenvalue of 1, and the corresponding eigenvector x_∞ is the stationary distribution of the Markov chain represented by A, i.e.

$$x_\infty = \lim_{m \to \infty} A^m x, \forall x$$

However, the above property does not hold if the underlying graph is not strongly connected. PageRank avoids this problem by computing x_∞ from a matrix $B = \lambda A + (1 - \lambda)C$, where $C_{ij} = 1/n$ and λ is a constant. C allows random jumps of equal probability between vertices, and guarantees the graph to be strongly connected. For plot graphs, we also insert an edge from every ending event to every starting event.

The intuition behind Personalized PageRank, and thus EventRank, is that the matrix C can be biased so the random jumps can favor some vertices for semantic reasons. EventRank incorporates the information of event cluster size and mutual exclusion from the plot graph into the transition matrix. We compute a matrix M as

$$M = \lambda A + (1 - \lambda)E$$

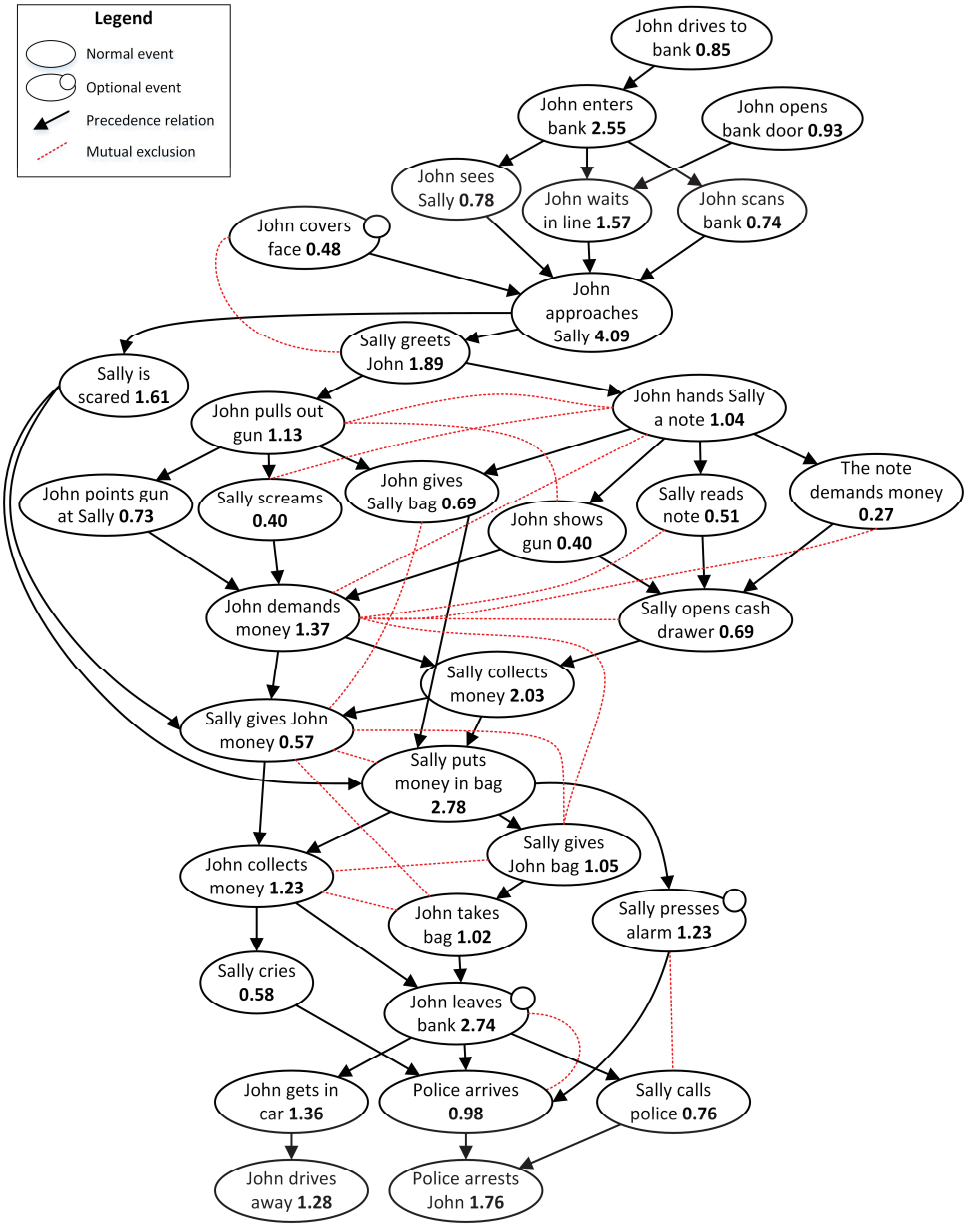

Figure 3. Importance of events in the bank robbery domain.

The matrix A is the same transition matrix computed from the edges, i.e. temporal orderings. λ is set to 0.7. The matrix E is the frequency that each event appears in the original corpus of exemplar stories with mutual exclusion relations factored in. We emphasize frequent events because they are probably more important than infrequent events for a given social situation. The frequency f_i of event i is just the number of times it is mentioned in all crowdsourced exemplar stories divided by the number of exemplar stories. The probability of randomly jumping from any event to i should be proportional to f_i. However, when event i and event j are mutually exclusive (denoted as $i \nsim j$), the transition from i to j should have a greatly reduced probability of occurring. Thus, we construct the matrix $\acute{E}$ as

$$\acute{E}_{ij} = \begin{cases} f_i - \frac{1}{2} o_j, & \text{if } i \nsim j \\ f_i, & \text{otherwise} \end{cases}$$

where

$$o_i = \sum_{j \notin X_i} f_j$$

and X_i is the set of vertices mutually exclusive to vertex i. The rationale is that if an event has fewer mutually exclusive events, it is more likely to be included in a story and hence more powerful in weakening other events. Finally, we normalize entries in $\acute{E}$:

$$E_{ij} = \frac{\acute{E}_{ij}}{\sum_{1 \le j \le n} \acute{E}_{ij}}$$

Event	Importance
John drives to bank	0.85
John opens bank door	0.93
John enters bank	**2.55**
John scans bank	0.74
John waits in line	**1.57**
John sees Sally	0.78
John approaches Sally	**4.09**
Sally greets John	**1.89**
John pulls out gun	1.13
John points gun at Sally	0.73
Sally screams	0.4
Sally is scared	**1.61**
John demands money	**1.37**
John gives Sally bag	0.69
Sally collects money	**2.03**
Sally puts money in bag	**2.78**
John collects money	1.23
Sally presses alarm	1.23
John leaves bank	**2.74**
Sally cries	0.58
Police arrests John	**1.76**

Figure 4. Events in a complete alibi story selected using importance. The 10 most important events are shown in bold and the 5 least important events are underlined.

so that each column sum up to 1. We again compute the stationary distribution by finding the eigenvector of M corresponding to the eigenvalue 1.

Figure 3 shows a plot graph in the bank robbery domain and the importance for all events. John is a bank robber who robbed a bank where Sally works. We can observe that events at bottleneck positions (e.g. "John approaches Sally") have large importance values, demonstrating the effect of graph structure on importance. Major events of two mutual exclusive branches (e.g. "John pulls out gun" and "John hands Sally a note") are of similar importance, reflecting the fact they have similar status.

We perform discourse planning based on the plot graph in Figure 3. A random walk of the plot graph produces a totally ordered event sequence shown in Figure 4 (the random walk algorithm is fully explained in [12]). Out of the 23 events, the 10 most important events (shown in bold) constitute a short summary including major events such as entering and leaving, demanding money and being arrested. If an NPC wants to provide minor but interesting detail, she can choose to add some of the least important events like Sally screaming and crying. A user study evaluating the degree that discourse planning agrees with human intuition is ongoing work.

3.2 Text Generation

After deciding on what events to tell, we produce the textual realization by selecting a sentence from each event cluster in the discourse plan. We consider two dimensions: (1) the interestingness of the text (different from interestingness of events discussed in the previous section) and (2) the sentiment in the text. Both aspects can reflect NPCs' personal traits and diversify the NPCs in terms of how they speak. Some NPCs may speak very succinctly with little interesting details, whereas others can recall vivid details. In addition, the NPC can describe the events with positive or negative sentiments. Given a linear sequence of events, where each event is a cluster of natural language sentences, natural language text is generated by selecting the sentence from each cluster that best matches the intended personality and sentiment based on these criteria. In a post-processing stage, the system rewrites sentences in which the agent is the actor to be first-person singular using a set of grammatical rules.

3.2.1 Textual Interestingness

We consider two aspects of language that could make storytelling interesting. The first is the extent to which sufficient details about events are provided. The second aspect is to describe these details with expressive language and with accurate descriptions of emotions and actions.

We first model the amount of details as the probability of a sentence in English, as Information Theory suggests a less likely sentence should contain more information. To this end, we utilize the Google N-Gram corpus, which aggregates the frequency of words and n-grams in books published from the 16th century to the present day. Due to its large size, the frequencies of words in this corpus approximate their probability in English. In the bag-of-word model, each word is independently drawn from a probability distribution over all English words. Thus, the probability of generating a particular sentence S containing words $\langle w_1, w_2, \ldots, w_k \rangle$ each appearing $\langle x_1, x_2, \ldots, x_k \rangle$ times follow from the multinomial distribution:

$$\mathrm{P}(S) = \frac{n!}{\prod_{i=1}^{k} x_i!} \prod_{i=1}^{k} \mathrm{P}(w_i)^{x_i}$$

where $n = \sum_{i=1}^{k} x_i$, and $\mathrm{P}(w_i)$ is the probability of word w_i. For our purpose, the average frequency over the 10-year period of 1991 to 2000 in the "English 2012" corpus is used. Stop words are removed before computation.

We further consider the style of language is as how much it resembles fictional novels. The language used in fictions may be distinguished from general English by word choice, such as vivid descriptions of actions (e.g. "snatch" instead of "take"), more emotional words, and less business-sounding words (e.g. facility, presentation). We can also obtain this information from the Google N-Gram corpus of fiction books. If a word appears more often in fiction books than in all books, we can presume that its use is more likely to create a sense that a story is being told in a more storyline fashion—more like literary text—than general text. Therefore, the *fictionality* of a word w can be computed as

$$f_w = \frac{\mathrm{P}_{fic}(w)}{\mathrm{P}_{all}(w)}$$

where $\mathrm{P}_{all}(w)$ is the probability of a word appearing in all books and $\mathrm{P}_{fic}(w)$ is the probabilities of a world appearing in fiction books (the "English Fiction 2012" corpus). We aggregate fictionality values of individual words in a sentence as an exponential average:

$$\mathrm{fic}(S) = \frac{\sum_{w \in W} \exp(\alpha f_w)}{\mathrm{card}(W)}$$

where W is the set of words in the sentence S and α is a parameter. The exponential function puts more weights on more fictional words so that a few highly fictional words are not cancelled off by a large number of words with low fictionality.

Example event 1: John covers face

- MP: John put on a fake mustache.
- LP: John kept his head down as he pulled open the outer door and slipped his Obama mask over his face.
- MF: John looked at his reflection in the glass of the door, gave himself a little smirk and covered his face.
- MID: John kept his head down as he pulled open the outer door and slipped his Obama mask over his face.

Example event 2: Sally puts money in bag

- MP: Sally put $1,000,000 in a bag.
- LP: Sally put the money in the bag, and collected the money from the 2 tellers next to her.
- MF: Sally quickly and nervously stuffed the money into the bag.
- MID: Sally quickly and nervously stuffed the money into the bag.

Example event 3: John drives away

- MP: John drove away.
- LP: John pulled out of the parking lot and accelerated, thinking over which route would make it easier to evade any police cars that might come along.
- MF: John sped away, hoping to get distance between him and the cops.
- MID: John sped away, hoping to get distance between him and the cops.

Figure 5. Sentences selected according to different metrics: Most probable (MP), least probable (LP), most fictional (MF), and most interesting details (MID).

For sentences in the same event cluster, we find the most fictional sentence often provides a more vivid description, and the least probable sentence contains more objective details. We combine these features using the harmonic mean of their ranks under the least probable metric and the most fictional metric: r_{LP} and r_{MF}. That is, the least probable sentence has $\mathrm{r}_{\mathrm{LP}} = 1$ and so on. The mean rank is:

$$\mathrm{r}_{\mathrm{MID}} = \frac{2\mathrm{r}_{\mathrm{LP}}\mathrm{r}_{\mathrm{MF}}}{\mathrm{r}_{\mathrm{LP}} + \mathrm{r}_{\mathrm{MF}}}$$

The sentence with the lowest $\mathrm{r}_{\mathrm{MID}}$ is picked as the sentence with the most interesting details. We find the most probable sentence usually provides a good summary for the event. Figure 5 shows sentences for some example events. Note the MF sentence usually contains more subjective emotions and character intentions, whereas the LP sentence is usually longer and contains more details. The MID sentence can be seen as a balance between the level of detail and the narrative language.

3.2.2 Textual Sentiments

Virtual characters may speak with the intention to express positive or negative sentiment. To detect sentiments of sentences in each event cluster, we construct a sentiment dictionary, called SentiWordNet+. SentiWordNet [1] is a sentiment dictionary that tags each *synset* (word sense) in WordNet [17] with three values: positivity, negativity, and objectiveness (objective words evoke no affect), all of which must sum to 1.0. Although SentiWordNet provides good coverage of words, we empirically find it to contain a large number of erroneous values, resulting in unreliable sentiment judgments on our event clusters. SentiWordNet+ is identical in nature to SentiWordNet—indeed we seed our dictionary with values from SentiWordNet—but uses an unsupervised, corpus-based machine learning technique to correct errors found in the original library. The intuition behind SentiWordNet+ is that words in the same neighborhood, including adjacent words and words in the same sentences and the same paragraph, should share similar sentiments, allowing us to automatically "smooth" any errors in the original sentiment library. In addition, words closer should have a stronger influence than words farther away.

We randomly selected 9108 books from Project Gutenberg that are written in English and tagged as fiction. The list of these books is at http://www.cc.gatech.edu/~bli46/SBG/list.txt. These books are tagged with parts of speech (POS) with the Stanford POS Tagger [28]. Each pair of word and POS is considered unique, so the same words with different POS are considered as different words.

For every occurrence of a target word we want to compute sentiment value for, we consider a neighborhood of 100 words, including 50 to the left and the right of the target word respectively. The word in the center of the neighborhood is at position 0. The word to the target's immediate left is at position -1, and the word to its immediate right is at position 1, and so forth. Only nouns, verbs, adjectives and adverbs in complete sentences in this neighborhood can influence the target word, and their positions are included in the index set $\mathcal{W}$. For a word w_i at position $i \in \mathcal{W}$, we place a Gaussian kernel function g_i centered at its position, which indicates the influence of word w_i on another word at position j:

$$g_i(j) = \exp\left(\frac{-(i-j)^2}{d}\right)$$

where d is a parameter deciding how fast the function diminishes with distance. We empirically set d to 32. The sentiment $s_{w_0}^k$ of the target word in the k^{th} neighborhood is computed as the weighted average for all kernel functions at position 0:

$$s_{w_0}^k = \frac{\sum_{i\in\mathcal{W}} s_{w_i}^{\mathrm{swn}}\, g_i(0)}{\sum_{i\in\mathcal{W}} g_i(0)}$$

where $s_{w_0}^{\mathrm{swn}}$ is the sentiment retrieved from SentiWordNet, i.e. the difference between the positive and negative sentiments for the word. The SentiWordNet value for the target word has no influence on the computed value $s_{w_0}^k$, i.e. $0 \notin \mathcal{W}$. The final sentiment value for the target word $\bar{s}_{w_0}$ is the average of all its occurrences in the corpus. We aggregate sentiments of individual words in sentence S, again using the exponential average:

$$\mathrm{senti}(S) = \frac{\sum_{i\in V} \mathrm{sign}\left(\bar{s}_{w_i}\right) \exp\left(\beta\left|\bar{s}_{w_i}\right|\right)}{\mathrm{card}(V)}$$

where $\mathrm{card}(V)$ is the cardinality of the index set V, which contains any noun, verb, adjective or adverb in that sentence. β is a scaling parameter. The exponential function ensures that words expressing strong sentiments are weighted more heavily than words with weak sentiments.

We selected a subset of English words that are of interests to our task. Crowdsourced stories in the bank robbery situation contain 504 unique nouns, verbs, adverbs and adjectives (i.e. the story words). We also selected some highly influential adjectives and adverbs that were direct neighbors of these story words. This gave us a total of 7559 words. After computing the raw sentiment values for these words, we normalize the values so that 1 percentile and 99 percentile of the values fall in the range of [-1, 1], in order to account for outliers. The dictionary can be downloaded from http://www.cc.gatech.edu/~bli46/SBG/dic.txt. β is set to 6 for the bank robbery situation.

Example event 1: Sally puts money in bag

- MP: Sally continued to cooperate, putting the money into the bag as ordered.
- MN: Sally's hands were trembling as she put the money in the bag.

Example event 2: Sally cries

- MP: Sally cried, somewhat relieved it may be over soon.
- MN: Sally felt tears streaming down her face as she let out sorrowful sobs.

Example event 3: Sally calls police

- MP: Sally described John as best as she could to the police.
- MN: Still shaken, Sally reached for the phone and in a panicked manner called the police.

Example event 4: John opens bank door

- MP: John took a deep breath and opened the bank door, letting an elderly woman exit before he entered himself.
- MN: John opened the bank door while his heart was beating fast.

Example event 5: John pulls out gun

- MP: John pulled out the gun, still smiling.
- MN: John reached behind his back and withdrew his pistol.

Figure 6. Sentences selected for most positive (MP) and most negative (MN) sentiments.

Figure 6 show some of the most positive (MP) and most negative (MN) sentences. We find the results tend to reflect the valences of individual words. In example events 1-3, individual words like "trembling" or "relieve" dominate the entire sentence, and we can correctly identify positive and negative sentences. In example event 4, "elderly" and "woman" have positive valences, which coincide with the semantic meaning of the sentence. However, there are also cases where the aggregation of individual words' valences deviates from the semantic meaning of the sentence. In example 5, the positive value of "smile" is the main reason for selecting the positive sentence, but smiling criminals may appear even scarier than usual.

3.2.3 Crowdsourcing Colorful Textual Descriptions

For the purpose of learning plot graphs, we asked crowd workers to write stories in simple and bland language [11]. Though simplified language facilitates plot graph learning by side-stepping many hard natural language processing problems, it is not conducive to generating vivid or sentimental speech. After learning the plot graph, we perform a second round of crowd-sourcing as an attempt to collect interesting event descriptions for each learned event cluster. The system recruits crowd workers on Amazon Mechanical Turk. Each worker is shown the events that constitute a complete story so that they understand the story context. For the compensation of $1, they are asked to write detailed descriptions for each of these events, and are hinted to describe characters' intentions, facial expressions and actions. Via this process, we collected 210 additional sentences. Many of these sentences can be seen in prior examples showing least probable and most fictional sentences for particular clusters (the most probably sentence typically comes from the original, simplified language exemplars)

On average, contributed "colorful" sentences have 2.6 verbs and are 13.7 words long, compared to the original corpus of exemplar stories which have 1.1 verbs and are 5.5 words long. Out of the 12 tasks we ran, we manually rejected 2 tasks because the sentences were just reworded with no more detail added or the sentences were not adhering to the story context.

4. DISCUSSION AND FUTURE WORK

Social believability is achieved by creating the appearance that an NPC understands and has experienced common social situations in the real world, despite the fact that an NPC has only ever lived in a paired-down virtual world designed for the express purpose of communicating with or entertaining the player. Social behaviors and social knowledge can be manually authored, as is often the case. As virtual worlds become richer and more expansive, human players' expectations of NPCs will rapidly outpace the ability to manually encode knowledge.

Our approach to social believability of NPCs is to incorporate data about the real world into the virtual characters. Crowdsourced corpora of social situations, the Google N-gram corpus, and Project Gutenberg corpus are all ways of providing an intelligent agent with observations about how the real world works and the language used by real humans. The advantage of this approach is that NPCs can be given vague specification about what they should talk about and how they should talk about it and intelligent algorithms can exploit these corpora to fill in the details automatically. Thus NPCs can theoretically discuss any topic in any style of discourse, with any type of sentimental inflection.

Section 3 demonstrates how a single personal trait can be mapped to the selection of events and sentences. Thus, we are capable of generating archetype NPCs, such as someone who always speaks negatively. We may also want to combine multiple personal traits. We discussed using the harmonic mean to combine different metrics, but this is by no means the only way. For instance, when an NPC is required to be happy, we may select only among the positive sentences. Picking out a single best approach requires careful consideration of specific needs of the application and empirical evaluation in the form of user studies. Mapping established psychological models, such as the Big Five model, to these personal traits obtained from data may also provide further insights and is left for future work.

Further work is also required to consider the connections between sentences selected from event clusters. The plot graph and alibi generation ensures that sentences that reflect the events make sense with respect to event ordering. However, details referenced across events are not checked for consistency. For example, one sentence in the bank robbery example can tell us the robber asked for one million dollars, and the next sentence describes the event of handing over $100,000. Solving this problem requires greater semantic understanding of sentences.

5. CONCLUSIONS

Socially believable non-player characters are required to maintain the illusion of leading an actual life in the virtual world. For this purpose, we propose an approach to generate background stories, or alibi stories, for NPCs that explain their daily activities when not in the presence of the player. We specifically discuss techniques for NPCs to tell these stories according to different personal traits, such as attention to detail, conciseness, vividness, and current sentiments. In developing these techniques, we propose EventRank, an algorithm for determining the importance of events in a plot graph. We build a sentiment dictionary SentiWordNet+ by correcting errors in automatically generated sentiment values in SentiWordNet and adapting these values in a storytelling setting.

Driven by data sets consisting of crowdsourced exemplar sentences for events, the Google N-Gram Corpus, and Project Gutenberg, our alibi telling techniques help to overcome the authoring bottleneck for socially believable NPCs and to reduce the author's subjectivity in creating character dialogues. The effort presented in this paper and in prior works [11, 12] moves the state of the art closer to the vision of social believability without manual knowledge engineering.

6. ACKNOWLEDGMENTS

We gratefully acknowledge DARPA for supporting this research under Grant D11AP00270. We thank Stephen Lee-Urban and Rania Hodhod for valuable inputs.

7. REFERENCES

[1] Baccianella, S., Esuli, A., Sebastani, F. 2010. SENTIWORDNET 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 7th conference on International Language Resources and Evaluation.*

[2] Bates, J. 1992. Virtual reality, art, and entertainment. *Presence: The Journal of Tele-operators and Virtual Environments*, 1, 133-138.

[3] Bickmore, T., and Schulman, D. 2009. A virtual laboratory for studying long-term relationships between humans and virtual agents. *Proceedings of the 8th AAMAS Conference*.

[4] Bickmore, B., Schulman, D., and Yin, L. 2009. Engagement vs. deceit: Virtual humans with human autobiographies. *Proceedings of the 2009 International Conference on Intelligent Virtual Agents*.

[5] Elliott, C. and Brzezinski, J. 1998. Autonomous agents as synthetic characters. *AI Magazine*, 19(2), 13-30.

[6] Gratch, J. and Marsella, S. 2004. A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, 5(4), 269-306.

[7] Gratch, J., Wang, N., Okhmatovskaia, A., Lamothe, F., Morales, M., Van Der Werf, R., and Morency, L.-P. 2007. Can virtual humans be more engaging than real ones? *Proceedings of the 12th International Conference on Human-Computer Interaction.*

[8] Haveliwala, T.H. 2002. Topic-Sensitive PageRank. *Proceedings of the 11th International World Wide Web Conference*.

[9] Hedlund, J., Antonakis, J., and Sternberg, R. 2002. *Tacit Knowledge and Practical Intelligence: Understanding the Lessons of Experience*. Technical Report. ARI Research Note 2003-04, United States Army Research Institute for the Behavioral and Social Sciences.

[10] Lester, J.C., Ha, E., Lee, S., Mott, B., Rowe, J., Sabourin, J. 2013. Serious games get smart: Intelligent game-based learning environments. *AI Magazine*, 34(4), 31-45.

[11] Li, B., Lee-Urban, S., Appling, D.S., and Riedl, M.O. 2012. Crowdsourcing narrative intelligence. *Advances in Cognitive Systems*, 2.

[12] Li, B., Lee-Urban, S., Johnston, G. and Riedl, M.O. 2013. Story generation with crowdsourced plot graphs. *Proceedings of the 27th AAAI Conference on Artificial Intelligence*.

[13] Lin, G. and Walker, M. 2011. All the world's a stage: Learning character models from film. *Proceedings of the 7th AAAI Conference on Artificial Intelligence for Interactive Digital Entertainment*.

[14] Lu, Y., Castellanos, M., Dayal, U., and Zhai, C.X. 2011. Automatic construction of a context-aware sentiment lexicon: An optimization approach. *Proceedings of the International World Wide Web Conference*.

[15] McCoy, J., Treanor, M., Samuel, B., Tearse, B., Mateas, M., and Wardrip-Fruin, N. 2010. Comme il Faut 2: A fully realized model for socially-oriented gameplay. *Proceedings of the 3rd Workshop on Intelligent Narrative Technologies.*

[16] Michel, J.-B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., Brockman, W., The Google Books Team, Pickett, J.P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., Pinker, S., Nowak, M.A., and Aiden, E.L. 2011. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*. 331. 176-182.

[17] Miller, G.A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39-41.

[18] Mohammad, S.M. and Turney, P.D. 2010. Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon. *Proceedings of the NAACL HLT Workshop on Computational Approaches to Analysis and Generation of Emotion in Text.*

[19] Mohammad, S.M. 2011. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. *In Proceedings of the ACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities.*

[20] Orkin J., and Roy, D. 2009. Automatic learning and generation of social behavior from collective human gameplay. *Proceedings of the 8th International Conference on Autonomous Agents and Multi Agent Systems*.

[21] Perrie, J., Islam, A., Milios, E., Keselj, V. 2013. Using Google n-grams to expand word-emotion association lexicon. *Proceedings of the 14th international conference on Computational Linguistics and Intelligent Text Processing.*

[22] Reeves, B., and Nass, C. 1996. *The Media Equation.* Cambridge, UK: Cambridge University Press.

[23] Robertson, J., and Young, R.M. 2013. Modelling Character Knowledge in Plan-Based Interactive Narrative to Extend Accomodative Mediation. *Proceedings of the Intelligent Narrative Technologies Workshop.*

[24] Sunshine-Hill, B., and Badler, N.I. 2010. Perceptually Realistic Behavior through Alibi Generation. *Proceedings of the Sixth AIIDE Conference.*

[25] Si, M., Marsella, S., and Pynadath, D. 2010, Modeling appraisal in theory of mind reasoning, *Journal of Autonomous Agents and Multi-Agent Systems*, 20, 14-31.

[26] Swanson, R. and Gordon, A. 2012. Say Anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems*, 2, 16:1-16:35.

[27] Swartout, W., Artstein, R., Forbell, E., Foutz, S., Lane, H.C., Lange, B., Morie, J., Rizzo, A., and Traum, D. 2013. Virtual humans for learning. *AI Magazine*, 34(4), 13-30.

[28] Toutanova, K., Klein, D., Manning, C., and Singer, Y. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL.*