Automatically Learning to Tell Stories about Social Situations from the Crowd

Boyang Li, Stephen Lee-Urban, Darren Scott Appling, and Mark O. Riedl
School of Interactive Computing; Georgia Institute of Technology
Atlanta, Georgia 30032 USA
E-mail: {boyangli, lee-urban, darren.scott.appling, riedl}@gatech.edu

Abstract

Narrative intelligence is the use of narrative to make sense of the world and to communicate with other people. The generation of stories involving social and cultural situations (eating at a restaurant, going on a date, etc.) requires an extensive amount of experiential knowledge. While this knowledge can be encoded in the form of scripts, schemas, or frames, the manual authoring of these knowledge structures presents a significant bottleneck in the creation of systems demonstrating narrative intelligence. In this paper we describe a technique for automatically learning robust, script-like knowledge from crowdsourced narratives. Crowdsourcing, the use of anonymous human workers, provides an opportunity for rapidly acquiring a corpus of highly specialized narratives about sociocultural situations. We describe a three-stage approach to script acquisition and learning. First, we query human workers to write natural language narrative examples of a given situation. Second, we learn the set of possible events that can occur in a situation by finding semantic similarities between the narrative examples. Third, we learn the relevance of any event to the situation and extract a probable temporal ordering between events. We describe how these scripts, which we call plot graphs, can be utilized to generate believable stories about social situations.

Introduction

Storytelling, in oral, visual, or written forms, plays a central role in various types of entertainment media, including novels, movies, television, and theatre. The prevalence of storytelling in human culture may be explained by the use of narrative as a cognitive tool for situated understanding (Bruner 1991; McKoon & Ratcliff 1992; Gerrig 1993; Graesser, Singer & Trabasso 1994). This *narrative intelligence* (Mateas & Sengers 1999) is central in the cognitive processes employed across a range of experiences, from entertainment to active learning. It follows that computational systems possessing narrative intelligence may be able to interact with human users naturally because they understand collaborative contexts as emerging narrative and are able to express themselves by telling stories.

In this paper we consider the problem of creating and telling stories that involve common social situations. Most stories are about people (or objects and animals that behave like people in some way). Characters in generated stories should respect social and cultural norms, and perform common tasks in socioculturally acceptable ways. For example, during a trip to a restaurant, a character should perform actions that meet readers' expectation of what should happen in a restaurant. Further, to generate a love story in which a boy asks a girl out to a date at the movies, a system should know when it is okay for the boy to hold the girl's hand or when to try for a kiss. To omit these elements or to use them at the wrong time invites failures in believability or breakdowns in communication.

The generation of believable stories requires extensive knowledge that captures common social and cultural activities. Unfortunately, social and cultural models are notoriously hard to model by hand. For example, a simple model of restaurant behaviour uses 87 rules (Mueller 2007). A simulation game about attending a prom (McCoy et al. 2010) required 5,000 rules to capture the social dynamics associated with that situation.

As an alternative to production rules, one may consider employing *scripts* (Schank and Abelson 1977), a form of procedural knowledge that describes how common situations are expected to unfold, thus capturing social and cultural norms. A script about visiting a restaurant, for example, would encode the typical progression of events (entering, being seated, reading a menu, paying the bill, etc.). Many story generation systems make use of manually coded script-like knowledge, such as cases or hierarchical task libraries (e.g. Meehan 1976; Lebowitz 1987; Turner 1994; Perez y Perez & Sharples 2001; Cavazza, Charles, & Mead 2002; Gervas et al. 2005; Swanson & Gordon 2008; Riedl 2010; Li & Riedl 2010; Hajarnis et al. 2011). However, the effort required to manually code script-like information becomes a significant bottleneck. As a result, most story generation systems to date are restricted to a small number of hand-authored knowledge structures and can thus only operate within the bounds of a limited micro-world for which knowledge has been provided.

Automatically acquiring sociocultural knowledge can open up story generation systems to a wider repertoire of possible stories and domains. In this paper, we propose an approach for learning script-like knowledge from *crowdsourced* narrative examples. *Crowdsourcing* replaces a dedicated expert who solves a complicated problem with many members of the general public, or *workers*, each solving a simple problem (cf. Howe 2006, Quinn & Bederson 2011). In our case, we request each worker to provide a short real-world example of a common situation for which we wish to learn a script. For example, we may ask workers to describe an experience of a restaurant visit. Workers then tell stories in natural language that include typical events for that situation. Crowdsourcing thus provides a means for rapidly acquiring a highly specialized corpus of

examples of a given situation, significantly simplifying the subsequent learning. Our initial results suggest that robust knowledge structures can be learned from small corpora containing only about 40 worker responses.

Our automated approach simultaneously learns both the events that comprise a situation and the typical ordering of these events from the crowdsourced narratives. By leveraging the crowd and its collective understanding of social constructs, we can learn a potentially unlimited range of scripts regarding how humans generally believe real-world situations unfold. We seek to apply this script-like knowledge to the generation of believable stories that involve common social situations or the direct engagement of virtual characters in social behaviors.

Background and Related Work

This section reviews story generation systems and discusses their reliance on hand-coded knowledge structures. We compare our crowdsourced approach for the acquisition of script-like knowledge to previous knowledge acquisition techniques and highlight its strengths and weaknesses.

Story Generation

Automated story generation systems search for a novel sequence of events that meet a given communicative objective, such as to entertain or convey a message or moral. The most common approaches to story generation are planning and case-based reasoning.

Planning-based story generation systems (Meehan 1971; Lebowitz 1987; Cavazza, Charles, & Mead 2002; Riedl & Young 2010; Li & Riedl 2010; Ware & Young 2011) use a causality-driven search to link a series of primitive actions to achieve a goal. The knowledge structures are usually too lean to fully represent common social scripts. Some story generation systems (cf., Lebowitz 1987; Cavazza, Charles, & Mead 2002; Li & Riedl 2010) attempt to enrich the generation process with hierarchical scripts that capture common ways of solving goals and performing tasks. System designers typically handcraft these hierarchical scripts.

Case-based story generators (Turner 1994; Perez y Perez & Sharples 2001; Gervas et al. 2005; Swanson & Gordon 2008; Riedl 2010; Hajarnis et al. 2011) attempt to construct novel stories by reusing prior stories, or cases. Sociocultural norms can be "baked into" the prior cases. Most case-based story generators to date have relied on hand-coded cases and stories, with two exceptions of note. First, the system described by Hajarnis et al. (2011) learns cases from human storytellers who enter stories via a custom interface. Cases can only be expressed in terms of a known set of possible actions, and are thus limited to a given micro-world. Second, SayAnything (Swanson & Gordon 2008) constructs new stories from fragments of stories mined from online blogs. This is a promising approach, although reliably selecting and reusing appropriate narrative fragments in the correct context remains an open problem. In contrast, our approach starts with a smaller number of crowdsourced stories specifically aimed at a particular situation that we wish to tell stories about, reducing the need to reason about context.

Script Knowledge Acquisition

Work on commonsense reasoning has sought to acquire propositional knowledge from a variety of sources. LifeNet (Singh & Williams 2003) is a commonsense knowledge base about everyday experiences constructed from 600,000 propositions asserted by the general public. According to Singh and Williams, this technique tends to yield spotty coverage. Gordon et al. (2011) describe an approach to mine causal relations from millions of blog stories. These systems do not attempt to create script-like knowledge representations; it is not clear how this knowledge would be used to generate novel stories. Open Mind Experiences (Singh & Barry 2003; Singh, Barry, & Liu 2004) is a database of stories and has been proposed as a means to generate new stories (Liu & Singh 2002).

Script-like knowledge can also be acquired from large-scale corpora with the goal of applying knowledge learned to the task of understanding news stories (Girju 2003; Bean & Riloff 2004; Brody 2007; Chambers & Jurafsky 2009; Kasch & Oates 2010). These systems attempt to find correlations between events appearing in these stories. In particular, the technique by Chambers and Jurafsky (2009) attempts to identify related event sentences and learn partially ordered *before* relations between events. While these works are intended to further natural language processing goals, such as script recognition, the learned scripts are general in nature and thus can be applied to a range of problems including story generation.

While corpus-based script learning can be very powerful, it also suffers from two limitations. First, the topic of the script to be learned must be represented in the corpus. Thus, it might be difficult to learn the script for how to go on a date to a movie theatre from a news article corpus. Second, given a topic, only the relevant events from the corpus should be extracted and irrelevant events should be excluded whereas a general corpus will have many irrelevant events that must be filtered. Ideally, one has a specialized corpus for each situation one wishes to learn a script for, but such specialized corpora rarely exist.

Crowdsourcing can be used to rapidly acquire a specialized corpus by paying, or otherwise incentivizing, a number of untrained human workers to provide examples of the topic in narrative form. With proper instructions, a crowd of amateurs can collectively create a specialized corpus from which high-quality scripts can be learned. The corpus will contain only relevant data and relatively complete examples of situations. In addition, the corpus may be specialized for any target domain. That is, crowdsourcing provides a means for rapidly acquiring a highly specialized corpus of examples of a given situation, which may significantly

simplify subsequent learning.

Crowdsourcing usually breaks up a complex problem into a number of simpler subproblems to make them easily solvable for ordinary workers. Hence, crowdsourced results must still be filtered, aggregated, and summarized in an automated fashion to create a complete solution. This collaborative human-AI approach has been used to train spell checkers (Lasecki et al. 2011), teach robots to perform tasks (Butterfield et al. 2010; Chernova, Orkin, and Breazeal 2010), construct learning materials (Boujarwah, Abowd, and Arriaga 2012), and tackle other challenging problems.

Jung et al. (2010) extract procedural knowledge from eHow.com and wikiHow.com where humans enter how-to instructions for a wide range of topics. Although these resources are sufficient for humans, for computational systems, the coverage of topics is sparse (very common situations are missing). Further, instructions in these websites tend to use complex language, conflate instructions and recommendations, and involve complex and nuanced conditionals.

In the *Restaurant Game,* Orkin and Roy (2009) use traces of people in a virtual restaurant to learn a probabilistic model of restaurant activity. *The Restaurant Game* as a playable interactive system has an *a priori* known set of actions that can occur in restaurants (e.g., sit down, order, etc.) that were programmed in advance. Users select actions to perform to recreate restaurant-going experiences, which the system then uses to learn probabilistic event ordering knowledge. Our work is similar to this, except our approach also learns the primitive events from natural language narrative texts, in addition to temporal orderings between events.

Crowdsourcing Narrative Examples

To learn a script for a particular, given situation we use a three-step process. First, we query crowd workers to provide linear, natural language narratives of the given situation. After some time, a small, highly specialized corpus of examples is acquired. Second, we identify the salient events in these narratives. This is in contrast with Orkin and Roy (2009), where the set of possible actions are known in advance. Third, we identify the order of these events. The second and third step work together to extract a script as a graph from the crowd-supplied narratives. As workers are not experts in knowledge representation, we do not ask workers to author script graphs directly; we believe that for lay workers, providing step-by-step narratives is a more intuitive and less error-prone means of conveying complex information than manipulating complex graphical structures.

In the crowdsourcing stage, to facilitate the subsequent learning of events and their ordering, our system includes precise instructions to the anonymous workers. First, we ask workers to use proper names for all the characters in the task. This allows us to avoid pronoun resolution problems. We provide a cast of characters for common roles, e.g., for the task of going to a fast-food restaurant, we provide named characters in the role of the restaurant-goer, the cashier, etc. Currently, these roles must be hand-specified, although we envision future work where the roles are extracted from online sources of general knowledge such as Wikipedia. Second, we ask workers to segment the narrative such that each sentence contains a single activity. Third, we ask workers to use simple natural language; specifically we ask them to use one verb per sentence and avoid using compound sentences. Throughout the remainder of the paper, we will refer to a segmented activity as a *step*. Figure 1 shows two fragments of narratives about the same situation.

Story A	**Story B**
a. John drives to the restaurant.	a. Mary looks at the menu.
b. John stands in line.	b. Mary decides what to order.
c. John orders food.	c. Mary orders a burger.
d. John waits for his food.	d. Mary finds a seat.
e. John sits down.	e. Mary eats her burger.
f. John eats the food.	…
…	

Figure 1. Example crowd-sourced narratives.

Once a corpus of narrative examples for a specific situation is collected from the crowd, we begin the task of learning a script. In our work, a script is a set of *before* relations, $B(e_1, e_2)$, between events e_1 and e_2 signifying that e_1 occurs before e_2. These relations coincide with causal and temporal precedence information, which are important for narrative comprehension (Graesser, Singer, and Trabasso 1994). A set of *before* relations allows for partial orderings, which can allow for variations in legal event sequences for the situation. The tasks of learning the main events that occur in the situation and learning the ordering of events are described in the next sections.

Event Learning

Event learning is a process of determining the primitive units of action to be included in the script. By working from natural language descriptions of situations, we learn the salient concepts used by a society to represent and reason about common situations. We must overcome several challenges:

1. The same step may be described in different ways.
2. Some steps may be omitted by some workers.
3. A task may be performed in different ways and therefore narratives may have different steps, or the same steps but in a different order.

Our approach is to automatically cluster steps from the narratives based on semantic similarity such that clusters come to represent the consensus events that should be part of the script. Each step in a narrative is a phrase that may or may not be semantically equivalent to another step in another narrative. There are many possible ways to cluster sentences based on semantic similarity; below we present the technique that leverages the simple language encouraged by our crowdsourcing technique. First, we preprocess the narratives to extract

the core components of each step: the main verb, the main actor, and the verb patient if any. Second, we identify the semantic similarity of each step using semantic gloss information from WordNet (Miller 1995). Finally we cluster steps in order to identify the core set of events.

Semantic Similarity

We use the Stanford parser (Klein & Manning 2003) to identify the actor, verb, and the most salient non-actor noun for each step. The most salient non-actor noun is identified using a rule-based approach. Once we have these components, the similarity between two corresponding components is computed as follows. For a pair of words (verbs or non-proper nouns), we obtain their similarity using the WordNet Gloss Vector technique (Patwardhan & Pedersen 2006). The WordNet Gloss Vector technique uses the cosine similarity metric to determine the similarity [0,1] for any two weighted term vectors for the desired synsets. To apply this technique, we need the appropriate WordNet synset for each verb or noun; we the Pedersen and Kolhatkar (2009) word-sense disambiguation technique to identify the best WordNet synset.

The similarity between two steps thus is computed as a weighted sum of the following elements:

- Semantic similarity of verbs
- Semantic similarity of nouns
- The difference in event location

Event location—a step's location as the percentage of the way through a narrative—helps disambiguate semantically similar steps that happen at different times, especially when a situation is highly linear with little variation. For example, when going to a movie theatre, one will "wait in line" to buy tickets and then may "wait in line" to buy popcorn. While both activities share semantic information, they should be considered distinct events.

Event Clustering

We model event learning as the clustering of steps, making use of the semantic information computed above. The resultant clusters are the events that can occur in the given situation.

Event clustering is performed in two stages. In the first stage, we make initial cluster assignments of steps from different narratives using shallow information. For each pair of steps from the same narrative, we record a *no-link* constraint, prohibiting these two steps from being placed into the same cluster. For each pair of steps from different narratives that have identical verbs and nouns, we record a *must-link* constraint, requiring that these two steps be placed within the same cluster. From this information, we produce an initial assignment of steps to clusters that respects all constraints.

In the second stage, we iteratively improve the cluster quality through the application of the k-Medoids clustering algorithm. The k-Medoids makes use of similarity between steps, as discussed above. We automatically set the similarity score to 1.0 if there is a *must-link* constraint between steps and 0.0 if there is a *no-link* constraint between steps.

The k-Medoid clustering algorithm requires k, the number of total clusters, to be known. We use a simple technique to sample different values for k, starting with the average narrative length, searching for a solution that minimizes intra-cluster variance while maximizing the extra-cluster distance.

Table 1. Crowd-sourced data sets.

Situation	Num. stories	Mean num. steps	Unique verbs	Unique nouns
Fast food	30	7.6	55	44
Movie date	38	10.7	71	84

Experiments and Results

To evaluate our event learning algorithm, we collected two sets of narratives for the following situations: going to a fast food restaurant, and taking a date to a movie theatre. While restaurant activity is a fairly standard situation for story understanding, the movie date situation is meant to be a more accurate test of the range of socio-cultural constructs that our system can learn. Table 1 shows the attributes of each specialized corpus.

For each situation, we manually created a gold standard set of clusters against which to calculate precision and recall. Table 2 presents the results of event learning on our two crowdsourced corpora, using the MUC6 cluster scoring metric (Vilain et al. 1995) to match computed cluster results against the gold standard. These values were obtained using parameter optimization to select the optimal weights for the clustering similarity function. The ideal weights for a given situation, naturally, depend on language usage and the degree to which variability in event ordering can occur. Table 2 shows how each portion of our algorithm helps to increase accuracy. Initial cluster seeding makes use of shallow constraint information. The semantic similarity columns show how phrase expansion improves our clusters. Event location further increases cluster accuracy by incorporating information contained in the implicit ordering of events from the example narratives. For each set of results, we show the average precision, recall, and F1 score for the best weightings for verb, noun, and event location similarity components.

Noting the differences between data sets, the movie

Table 2. Precision, Recall, and F1 Scores for the restaurant and movie data sets.

Situation	Gold std. num. events	Initial seed clusters			Semantic similarity			Semantics + Location		
		Pre.	Recall	F1	Pre.	Recall	F1	Pre.	Recall	F1
Fast food restaurant	21	0.780	0.700	0.738	0.806	0.725	0.763	**0.814**	**0.739**	**0.775**
Movie theatre date	56	0.580	0.475	0.522	0.725	0.580	0.645	**0.763**	**0.611**	**0.679**

date corpus has a significantly greater number of unique verbs and nouns, longer narratives, and greater usage of colloquial language. Interestingly, the movie date corpus contains a number of non-prototypical events about social interactions (e.g., *Sally slaps John.*) that appear rarely. This greater number of clusters containing few steps has a negative effect on recall values; a larger number of narratives would ameliorate this effect by providing more examples of rare steps. By crowdsourcing a highly specialized corpus, we are able to maintain precision in the face of a more complicated situation without restricting worker ability to express their conception of the salient points of the situation.

Improving Event Clustering with Crowdsourcing

While we believe that our event learning process achieves acceptably high accuracy rates, errors in event clustering may impact overall script learning performance (the effects of clustering errors on script learning will be discussed in a later section). To improve event-clustering accuracy, we can adopt a technique to improve cluster quality using a second round of crowdsourcing, similar to that proposed by Boujarwah, Abowd, and Arriaga (2012). Workers are tasked with inspecting the members of a cluster and marking those that do not belong. If there is sufficient agreement about a particular step, it is removed from the cluster. A second round of crowdsourcing is used to task workers to identify which cluster these "un-clustered" steps should be placed into. According to Boujarwah (personal communication), the multiple rounds of crowdsourcing required $110 for a single script, linearly increasing with situation complexity. Crowdsourcing is often used to improve on artificial intelligence results (von Ahn 2005) and we can increase clustering accuracy to near perfect in this way. However, in the long term our goal is minimize the use of the crowd so as to speed up script acquisition and reduce costs.

Plot Graph Learning

Once we have the events, the next stage is to learn the script structure. Following Chambers and Jurafsky (2009) we learn *before* relations $B(e_1, e_2)$ between all pairs of events e_1 and e_2. See Figure 2 for a visualization of a script as a graph. Chambers and Jurafsky train their system on the Timebank corpus (Pustejovsky et al. 2003), which uses temporal signal words. Girju (2003) uses causal signal words. Because we are able to leverage a highly specialized corpus of narrative examples of the desired situation, we can avoid reliance on signal words and instead probabilistically determine ordering relations between events directly from the narrative examples. The result of this process is a script-like structure similar in nature to a *plot graph* (Weyhrauch 1997), a partial ordering of events that defines a space of possible event sequences that can unfold during a given situation. Not only is a plot graph similar to a script, but it is also a data structure that has been used for AI story generation (Weyhrauch 1997; Nelson & Mateas 2005; Roberts et al. 2006; Sharma et al. 2010).

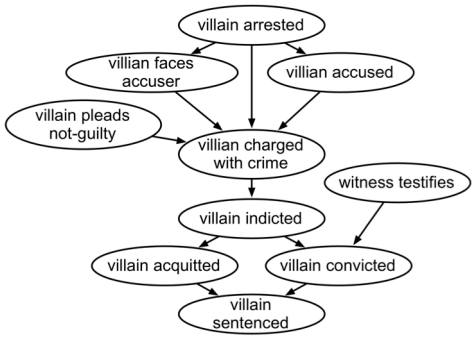

Figure 2. An example plot graph, adapted from Chambers and Jurafsky (2009).

Initial Script Construction

Script construction is the process of identifying the plot graph that most accurately captures the most information out of the set of crowdsourced narratives. Each possible *before* relation between a pair of events is a hypothesis (i.e. $B(e_1, e_2) = true$ or $B(e_2, e_1) = true$) that must be verified. For every pair of events e_1 and e_2, we count the observation of evidence for and against each hypothesis. Let s_1 be a step in the cluster representing event e_1, and let s_2 be a step in the cluster representing event e_2. If s_1 and s_2 appear in the same input narrative, and if s_1 appears before s_2 in the narrative, then we consider this as an observation in support of $B(e_1, e_2) = true$. If s_2 appears before s_1 in the same narrative, this observation supports $B(e_2, e_1) = true$.

The probability p_h of a hypothesis h equals k/n, where n is the number of observations and k is the observations that support h. Considering that the probability is only an estimate of the real world based on limited observations, we also estimate its confidence (cf. Wang 2009); a probability computed based on a small number of observations has low confidence. Without assuming prior distributions for orderings between arbitrary events, we use the imprecise Dirichlet model (Walley 1996) to represent this uncertainty. Suppose we have s additional observations whose values are hidden, the most optimistic estimate of the probability occurs when all hidden observations support hypothesis h, yielding an upper bound $p_h^+ = (k+s)/(n+s)$. Similarly, the most pessimistic estimate is $p_h^- = k/(n+s)$. Thus, the confidence in a probability is $c_h = 1 - \left(p_h^+ - p_h^-\right) = 1 - s/(n+s)$, where s is a parameter

We select relations for the plot graph in which the probability and confidence exceed thresholds $T_p, T_c \in [0,1]$, respectively. T_p and T_c apply to the entire graph and provide an initial estimate of the best plot graph. However, a graph that better explains the crowdsourced narratives may be found if the thresholds could be locally relaxed for particular relations. Below, we introduce a measure of plot graph error and an

Q := all of events (e_1, e_2) where e_2 is reachable from e_1 or unordered
Foreach (e_1, e_2) $\in$ ***Q*** in order of decreasing $D_N(e_1, e_2) - D_G(e_1, e_2)$ **do:**
  E := all events such that for each $e_i \in$ ***E***, $D_G(e_1, e_i) = D_N(e_1, e_2) - 1$
  Foreach $e_i \in$ ***E*** **do:**
    If edge $e_i \rightarrow e_2$ has probability and confidence less than T_p, T_c and will not create a cycle if added to the graph **do:**
      Strengthen the edge by adding one observation in support of it
      If $e_i \rightarrow e_2$ has probability and confidence greater than T_p, T_c and adding $e_i \rightarrow e_2$ to the graph decreases *MSGE* **do:**
        Add $e_i \rightarrow e_2$ to the graph
Return graph

Figure 3. The plot graph improvement algorithm.

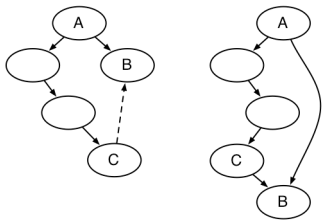

Figure 4. Compensation for errors between pairs of events.

algorithm for iteratively improving the plot graph to minimize the error.

Plot Graph Improvement

Since a plot graph encodes event ordering, we introduce an error measure based on the expected number of interstitial events between any pair of events. The error is the difference between two distance measures, $D_G(e_1, e_2)$ and $D_N(e_1, e_2)$. $D_G(e_1, e_2)$ is the number of events on the shortest path from e_1 to e_2 on the graph (e_1 excluded); this is also the minimum number of events that must occur between e_1 and e_2 in all legal totally ordered sequences consistent with the *before* relations of the plot graph. In contrast, $D_N(e_1, e_2)$ is the normative distance from e_1 to e_2 averaged over the entire set of narratives. For each input narrative that includes sentence s_1 from the cluster representing e_1 and sentence s_2 from the cluster representing e_2, the distance (i.e. number of interstitial sentences plus one) between s_1 and s_2 is $d_N(s_1, s_2)$. $D_N(e_1, e_2)$ is thus the average of $d_N(s_1, s_2)$ over all such input narratives. The mean squared graph error (MSGE) for the entire graph is:

$$MSGE = \frac{1}{|P|} \sum_{e_1, e_2 \in P} \left(D_G(e_1, e_2) - D_N(e_1, e_2) \right)^2$$

where P is the set of all ordered event pairs (e_1, e_2) such that e_2 is reachable from e_1 or that they are unordered.

We utilize this error measure to improve the graph based on the belief that D_N represents the normative distance we expect between events in any narrative accepted by the plot graph. That is, typical event sequences in the space of narratives described by the plot graph should have $D_G(e_1, e_2) \approx D_N(e_1, e_2)$ for all events. A particularly large $|D_N(e_1, e_2) - D_G(e_1, e_2)|$ may indicate that some edges with low probability or confidence could be included in the graph to make it closer to user inputs and reduce the overall error.

We implement a greedy, iterative improvement search for a plot graph that reduces mean square graph error (Figure 3). For each pair of events (e_1, e_2) such that e_2 is reachable from e_1 in the plot graph of directed edges, we search for all events E such that if $e_i \in E$ were the immediate predecessor of e_2 then $D_G(e_1, e_2)$ would be equal to $D_N(e_1, e_2)$. If there is a possible edge from e_i to e_2 (i.e., at least one observation that supports such an edge) then we strengthen the edge hypothesis by one observation. This intuition is illustrated in Figure 4 where the edge (dashed arrow) from event C to event B was originally insufficiently supported; adding the edge to the graph creates the desired separation between events A and B. This process repeats until no new changes to graph structure can be made that reduce the mean square graph error.

We find this approach to be effective at reducing graph error when T_p is set relatively high (> 0.5) and $T_c \approx 0.4$. A conservative T_p initially discards many edges in favor of a more compact graph with many unordered events. A moderate T_c allows the improvement algorithm to opportunistically restore edges to the graph.

Table 3. Error reduction for both situations.

Situation	Error before Improvement		Error after Improvement		Avg. Error Reduction
	Avg.	Min.	Avg.	Min.	
Fast food	4.05	1.23	2.31	0.85	42%
Movie date	6.32	2.64	2.99	1.88	47%

Experiments and Results

Figure 5 shows plot graphs learned for the fast food restaurant and movie theatre date situations. These plots were learned from the gold standard clusters under the assumption that we can achieve near perfect clustering accuracy with a second round of crowdsourcing. The event labels are English interpretations of each event based on manual inspection of the sentences in each event. For clarity, some edges are omitted from the figure that do not affect the partial ordering. Rare events, such as *Sally slaps John* are excluded from the graphs because their clusters contain too few sentences and thus do not meet our probability and confidence thresholds.

Some statistics about the two graphs are shown in Table 3. Over 128 sets of different parameter settings, we found that iterative graph improvement led to an average error reduction of 42% and 47% for the fast-food restaurant and movie data situations respectively. The asterisks in Figure 5 indicate edges that were added during graph improvement. Note that it is not always possible to reduce graph errors to zero when there are plausible ordering varations between events. For example *choose menu item* and *wait in line* can happen in any order, introducing a systematic bias for any graph path across this pair. In general we tend to see ordered relations when we expect causal necessity, and we see unordered events when ordering variations are supported by the data.

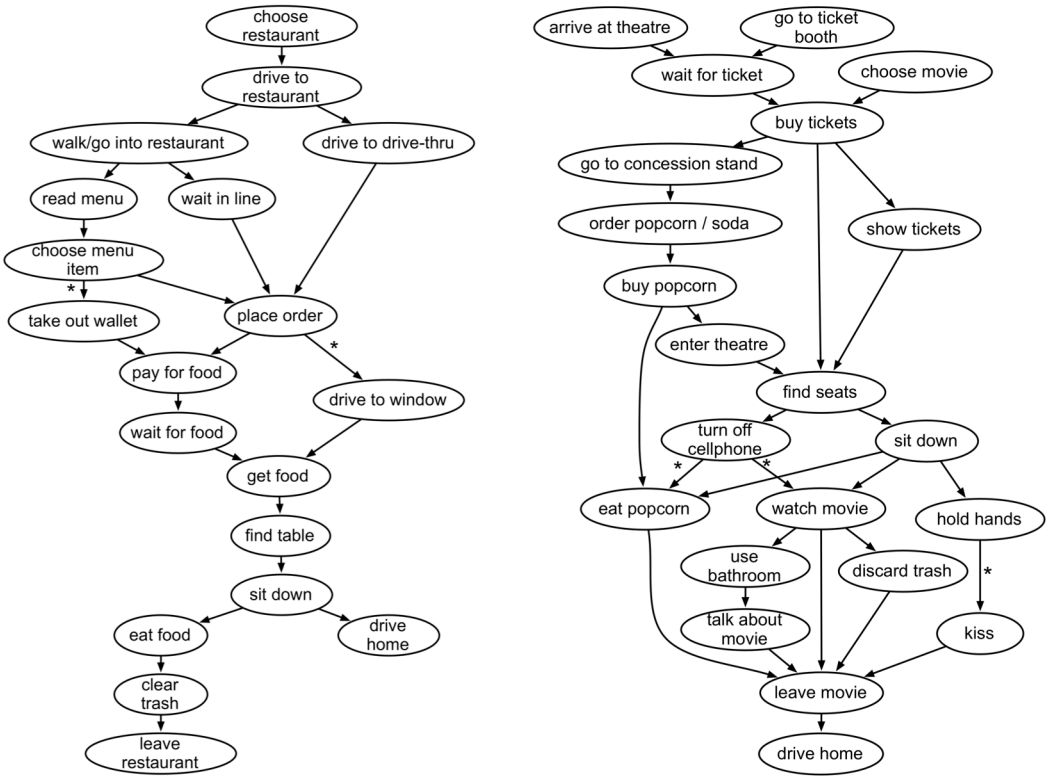

Figure 5. Plot graphs generated for the restaurant situation (left) and movie date situation (right).

Discussion and Future Work

There are several ways in which errors during event learning (i.e., clustering) can impact plot graph generation. First, steps may be improperly clustered, thus introducing observations of ordering relations between otherwise unrelated events, possibly causing cycles in the plot graph. If the number of improperly clustered sentences is relatively small, these relations have low probability and confidence and will be filtered out. Second, two distinct events may be merged into one event, causing ordering cycles in which all edges have high probability and confidence. When this happens, it is possible to eliminate the cycle by choosing an event to split into two. We select the event cluster in the cycle with the highest inter-cluster variance in the belief that high inter-cluster variance indicates that there is a natural split of sentences into two clusters. Third, an event may be split into two clusters unordered relative to each other. This creates the appearance that an event must occur twice in any story generated from this script.

Closely inspecting Figure 5, we note that *before* relations includes causal sufficiency and mere temporal precedence as well as strict causal necessity. For example, before *placing an order* at a fast-food restaurant one can *wait in line* or *drive to drive-thru* but not both. Thus, both are sufficient for *placing an order*. The crowdsourced corpus for the restaurant is split relatively evenly between walk-in and drive-thru narratives, implying two main variations to the situation (this also accounts for the unordered *leave restaurant* and *drive home* events). Future work will be necessary to distinguish causal and temporal relations as well as necessity versus sufficiency. We believe this can be accomplished by more fully leveraging correlations (e.g. mutual information) between events. As with the event learning phase, it is always possible to ask crowd workers to provide causal information with questions about causal counterfactuals, a technique adapted from Trabasso and Sperry (1985).

Toward Story Generation

To an extent, the plot graph learned as described above grants narrative intelligence to a computational process. A model of common social situations—in the form of a plot graph—captures common beliefs of how those real-world situations unfold. A computational system must also be able to act on this narrative intelligence in order to: (a) tell a story about a sociocultural situation, (b) tell a story in which a common social situation occurs, or (c) directly engage in a social situation in a virtual world. Fortunately, the plot graph representation facilitates story generation and interactive execution (cf., Weyhrauch 1997; Nelson & Mateas 2005; Sharma et al. 2010; Roberts et al. 2006).

A plot graph defines a space of totally ordered event sequences that are believed to be "legal" ways for a given situation to unfold. By virtue of the way we learn the plot graph from human-provided examples, the knowledge structure generalizes across the most common ways in which the given situation manifests. Within the space of legal stories, we may consider different possible storytelling goals: the most prototypical story, the most unusual story, the most

surprising, etc. According to Bruner (1991), interesting stories are those that deviate from the norm in some way.

The plot graph representation was originally used to determine what was possible for a user to do in an interactive fiction game. Although these systems are meant to provide narrative structure to games, we can view these systems as story generation systems when the interactive component is removed. To generate a story using a plot graph, a system must search for and select one totally ordered sequence from this set (Weyhrauch 1997; Nelson & Mateas 2005; Roberts et al. 2006; Sharma et al. 2010). To date, algorithms that use plot graphs have used the same set of heuristics to find sequences that reduce cognitive burden and reduce flailing, including:

- Location flow—events in the same location should occur together.
- Thought flow—events that are conceptually related should occur together.
- Motivation—a measure of whether plot points are motivated by previous plot points.

Other heuristic functions are used as well.

Generation of stories from *learned* plot graphs requires a slightly different approach. The plot graph describes a social situation that is relatively well constrained, so the only question that remains is how prototypical should the resultant story be. We define *typicality* as a function of the likelihood of events (nodes) and of specific sub-sequences (node-link-node sequences). By varying the inclusion of nodes and links according to their likelihood while respecting the before relations, we can generate stories that are legal but with arbitrary typicality within the norm.

We have a wealth of probabilistic information to draw from as a consequence of how we learn the plot graph, including:

- Typicality of events—the probability of an event being part of a situation, $P(e)$.
- Typicality of event orderings—the probability that a given ordering occurs, $P(e_1 \rightarrow e_2 \mid e_1 \wedge e_2)$.
- Adjacency—the probability that two events should occur immediately adjacent to each other, $P(e_1 * e_2 \mid e_1 \wedge e_2)$.
- Co-occurrence—the probability that any two events have been observed in the same crowdsourced story, $P(e_1 \wedge e_2)$.

The most prototypical story that can be generated from a given plot graph, for example, may be defined as inclusion of the n most probable events, ordered according to the most probable before relations between those n nodes. We can generate more interesting stories about the same situation by finding a legal sequence with (a) an unlikely event, such as kissing (kissing occurs in ~10% of crowdsourced examples); (b) likely events that occur in an unlikely ordering; (c) non-adjacent events that are typically adjacent; (d) pairs of events that have low co-occurrence; or (e) omission of an event that frequently co-occurs with a present event. We intend to investigate the effects of each of the above hypotheses on story novelty in order to develop tunable heuristics for the generation process.

Story generation from sociocultural plot graphs reaches full expressivity once we are able to differentiate links in the graph as denoting causal necessity or simple temporal precedence; this provides the richest variation among legal stories from which to choose a specific story or guide a virtual character's behavior. Once we differentiate between causal necessity and precedence, the story generation process can be performed using standard search techniques such as A*, forward or backward search, genetic algorithms and Monte Carlo methods.

Conclusions

Crowdsourcing provides direct access to humans and the ways in which they express experiential knowledge. A crowdsourcing approach has advantages over general corpus based learning: filtering irrelevant information, segmentation, and control of natural language complexity. Our approach capitalizes on these advantages by learning the primitive events from the segmented natural language and learning ordering constraints on these events directly from the crowd-sourced narrative examples.

Plot graph learning overcomes one of the primary bottlenecks in acquiring sociocultural knowledge required for effective generation of believable stories. While future work remains to tease out the full expressive power of automatically learned plot graphs, our approach makes it possible for a computational system to extend its narrative intelligence beyond a single, hand-crafted micro-world.

One of the strengths of our approach is the way in which we can leverage shared social constructs acquired directly from humans. Our approach learns the events that make up common situations directly from the language people use to describe those situations; event ordering captures shared social and cultural understanding based on people's descriptions of experiences. Thus, in addition to learning scripts for story generation, our system also learns a functional form of socio-cultural knowledge that could be applied to other computational narrative intelligence tasks such as story understanding.

Believable story generation requires in-depth understanding of the rich social situations that humans recognize and participate in everyday, yet this sort of experiential knowledge is rarely possessed by intelligent computational systems. A human-AI collaborative approach in which humans naturally convey experiential, social, and cultural knowledge to an intelligent system can overcome many of the hurdles to human-level AI problems.

Acknowledgements

The authors gratefully acknowledge the support of the U.S. Defense Advanced Research Projects Agency (DARPA) for this effort.

References

Bean, D., Riloff, E. (2004). Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of the 2004 HLT/NAACL Conference*.

Boujarwah, F., Abowd, G., Arriaga, R. (2012). Socially computed scripts to support social problem solving skills. In *Proceedings of the 2012 Conference on Human Factors in Computing Systems*.

Brody, S. (2007). Clustering clauses for high-level relation detection: an information-theoretic approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Bruner, J. (1991). The narrative construction of reality. *Critical Inquiry*, 18, pp. 1-21.

Butterfield, J., Osentoski, S., Jay, G., Jenkins, O.C. (2010). Learning from demonstration using a multi-valued function regressor for time-series data. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*.

Cavazza, M., Charles, F., Mead, S. (2001). Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation*, 13, pp. 121-131.

Chambers, N., Jurafsky, D. (2009). Unsupervised learning of narrative event chains. In *Proceedings of ACL/HLT 2009*.

Chernova, S., Orkin, J., Breazeal, C. (2010). Crowdsourcing HRI through online multi-player games. In *Proceedings of the 2010 AAAI Fall Symposium on Dialog with Robots*.

Gerrig, R. (1993). *Experiencing Narrative Worlds: On the Psychological Activities of Reading*. Yale University Press.

Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R. (2005). Story Plot Generation based on CBR. *Journal of Knowledge-Based Systems*, 18, pp. 235-242.

Girju, R. 2003. Automatic Detection of Causal Relations for Question Answering. *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering—Machine Learning and Beyond*.

Gordon, A., Bejan, C.A., Sagae, K. (2011). Commonsense causal reasoning using millions of personal stories. In *Proceeding of the 25th Conference on Artificial Intelligence*.

Graesser, A., Singer, M., Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review*, 101, pp. 371-395.

Hajarnis, S., Leber, C., Ai, H., Riedl, M.O., Ram, A. (2011). A case based planning approach for dialogue generation in digital movie design. In *Proceedings of the 19th International Conference on Case Based Reasoning*.

Howe, J. (2006). The rise of crowdsourcing. *Wired Magazine*, 14.06, June 2006.

Jung, Y., Ryu, J., Kim, K.-M., Myaeng, S.-H. (2010). Automatic Construction of a Large-Scale Situation Ontology by Mining How-to Instructions from the Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(2-3), pp. 110-124.

Kasch, N., Oates, T. (2010). Mining script-like structures from the web. In *Proceedings of the NAACL/HLT 2010 Workshop on Formailism and Methodology for Learning by Reading*.

Klein, D., Manning, C. (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Lasecki, W.S., Murray, K.I., White, S., Miller, R.C., Bingham, J.P. (2011). Real-time crowd control of existing interfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology*.

Lebowitz, M. (1987). Planning stories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*.

Li, B., Riedl, M.O. (2010). An offline planning approach to game plotline adaptation. In Proceedings of the 6th Conference on Artificial Intelligence *and* Interactive Digital Entertainment.

Liu, H., Singh, P. (2002). MAKEBELIEVE: using commonsense knowledge to generate stories. In *Proceedings of the 18th National Conference on Artificial Intelligence*.

Mateas, M., Senger, P. (1999). Narrative intelligence. In *Proceedings of the 1999 AAAI Fall Symposium on Narrative Intelligence*.

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

McKoon, G., Ratcliff, R. (1992). Inference during reading. *Psychological Review*, 99, pp. 440-466.

Meehan, J (1976). *The Metanovel: Writing Stories by Computers*. Ph.D. Dissertation, Yale University.

Miller, G. (1995). WordNet: a lexical database for english. *Communications of the ACM*, 38(11), pp. 39-41.

Mueller, E.T. (2007). Modelling space and time in narratives about restaurants. *Literary and Linguistic Computing*, 22(1), pp. 67-84.

Nelson, M., Mateas, M. (2005). Search-based drama management in the interactive fiction Anchorhead. In *Proceedings of the 1st Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Orkin J., Roy, D. (2009). Automatic learning and generation of social behavior from collective human gameplay. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*.

Patwardhan, S., Pedersen, T. (2006). Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL Workshop on Making Sense of Sense*.

Pedersen, T., Kolhatkar, V. (2009) WordNet:: SenseRelate::AllWords – a broad coverage word sense

tagger that maximimizes semantic relatedness. In *Proceedings of the ACL 2009 Conference*.

Pérez y Pérez, R., Sharples, M. (2001). MEXICA: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13, pp. 119-139.

Pustejovsky, J., Hanks, P. Saurí, R., See, A., Gaizauskas, R., Setzer, A. Radev, D. Sundheim, B., Day, D. Ferro, L., Lazo, M. (2003). The TIMEBANK Corpus. In *Proceedings of Corpus Linguistics*.

Quinn, A.J., Bederson, B.B. (2011). Human Computation: A Survey and Taxonomy of a Growing Field. In *Proceedings of The ACM SIGCHI Conference on Human Factors in Computing Systems.*

Riedl, M.O. (2010). Case-based story planning: creativity through exploration, retrieval, and analogical transformation. *Minds and Machines*, 20.

Riedl, M.O., Young, R.M. (2010). Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research*, 39, pp. 217-268.

Roberts, D.L., Nelson, M.J., Isbell, C.L., Mateas, M., Littman, M.L. (2006). Targeting specific distributions of trajectories in MDPs. In *Proceedings of the 21st National Conference on Artificial Intelligence*.

Schank, R., Abelson, R. (1977). *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates.

Sharma, M., Ontañón, S., Mehta, M., Ram, A. (2010). Drama Management and Player Modeling for Interactive Fiction Games. *Computational Intelligence*, 26, pp. 183-211.

Singh, P., Barry, B. (2003). Collecting commonsense experiences. In *Proceedings of the 2nd International Conference on Knowledge Capture*.

Singh, P., Barry, B, Liu, H. (2004). Teaching machines about everyday life. *BT Technology Journal*, 22(4), pp. 227-240.

Singh, P., Williams, W. (2003). LifeNet: a propositional model of ordinary human activity. In *Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture*.

Swanson, R., Gordon, A. (2008). Say Anything: a massively collaborative open domain story writing companion. In *Proceedings of the 1st International Conference on Interactive Digital Storytelling*.

Trabasso, T. and Sperry, L. (1985). Causal relatedness and importance of story events. Journal of Memory and Language, 24:595-611.

Turner, S. (1994). *The Creative Process: A Computer Model of Storytelling*. Lawrence Erlbaum Associates.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., Hirschman, L. (1995). A Model-Theoretic Coreference Scoring Scheme. In *Proceeding of the 6th Conference on Message Understanding* (MUC6).

von Ahn, L. (2005). *Human Computation*. Ph.D. Dissertation, Carnegie Mellon University.

Walley, P. (1996). Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58(1), pp. 3-57.

Wang, P. (2009). Formalization of evidence: a comparative study. *Journal of Artificial General Intelligence*, 1, pp. 25-53.

Ware, S., Young, R.M. (2011). CPOCL: a narrative planner supporting conflict. In *Proceedings of the 7th Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Weyhrauch, P. (1997). *Guiding Interactive Fiction*. Ph.D. Dissertation, Carnegie Mellon University.