Learning Sociocultural Knowledge via Crowdsourced Examples

Boyang Li, Darren Scott Appling, Stephen Lee-Urban, Mark O. Riedl

College of Computing, Georgia Institute of Technology, Atlanta, Georgia, USA
{ boyangli, darren.scott.appling, lee-urban, riedl}@gatech.edu

Abstract

Computational systems can use sociocultural knowledge to understand human behavior and interact with humans in more natural ways. However, such systems are limited by their reliance on hand-authored sociocultural knowledge and models. We introduce an approach to automatically learn robust, script-like sociocultural knowledge from crowd-sourced narratives. Crowdsourcing, the use of anonymous human workers, provides an opportunity for rapidly acquiring a corpus of examples of situations that are highly specialized for our purpose yet sufficiently varied, from which we can learn a versatile script. We describe a semi-automated process by which we query human workers to write natural language narrative examples of a given situation and learn the set of events that can occur and the typical even ordering.

Introduction

One of the major limitations to computational systems is their lack of sociocultural knowledge. Instilling computational systems will enable them to comprehend human behaviors and interact with humans in more natural ways. 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. We propose that virtual autonomous agents acquire sociocultural knowledge automatically. Today, virtual agents reside in a rich information ecosphere that contains a wealth of information, including the World Wide Web and human beings via means of crowdsourcing.

Crowdsourcing is the outsourcing of complicated tasks—typically tasks that cannot be performed by artificial intelligence algorithms—to a large number of anonymous workers via Web services (Howe 2006, Quinn & Bederson 2011). Crowds can be used to collect knowledge about how the world works. In this paper, we demonstrate a technique by which a computational system can deputize a crowd to provide sociocultural knowledge in a format that is easy to computationally consume. We propose a set of requirements that a solution to crowdsourcing sociocultural knowledge must meet:

1. Cost-effective. The time and cost of knowledge acquisition should be minimized by automating the construction of knowledge structures.
2. Natural crowd interactions. We should allow inputs in natural language. Processing natural language can be simplified by instructing workers to use restricted vocabulary and simplified language structures.
3. Capture situation variations. There may be variations and alternatives to common sociocultural situations.
4. Robust. Workers may skip steps, use ambiguous language, or describe events outside the scope of the situation. The system should reason about uncertainty and remain robust under noisy inputs.
5. Proactive. Knowledge learned from a crowd may not be complete. The system should identify opportunities for active learning or disambiguation and use additional rounds of crowdsoucing to improve results.

We automatically learn narrative *scripts* for a given sociocultural situation, such as going to a restaurant or going on a date to a movie theatre. Scripts (Schank and Abelson 1977) are procedural knowledge structures that encode the events that can occur during a situation and the expected ordering of events. While scripts could be constructed completely through human computation, we prefer to automate as much of the script acquisition from example narratives because (a) providing linear narratives is intuitively simpler than manipulating complex knowledge structures, and (b) narratives are natural means of conveying the social and cultural knowledge we wish to acquire. Our algorithms are designed to take into account the variability in worker narratives, attempting to distinguish between unreliable narrative accounts from natural variations in the situation. 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.

Related Work

Recent 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. However, this technique tends to yield spotty coverage (Singh and Williams 2003). Gordon et al. (2011) describe an approach to mining causal relations from millions of blog stories. While commonsense reasoning is related to automated story understanding, these systems do not attempt to create script-like knowledge representations.

There has been interest in learning script-like knowledge from large-scale corpora such as news corpora and other online sources of textual information (Bean and Riloff 2004; Brody 2007; Chambers and Jurafsky 2009; Girju 2003; Kasch and Oates 2010). Unlike other natural language processing techniques that learn correlations between sentences, these systems attempt to find relationships between many events. In particular, the technique by Chambers and Jurafsky (2009) attempts to identify related event sentences and learn partially ordered *before* relations between events.

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, one would not expect to learn the script for how to go on a date to a movie theatre from a news article corpus. Second, given a topic, a system must determine which events are relevant to the script. Given a corpus with a sufficient number of examples of the topic, a system can eventually determine relevance of any event with confidence. Ideally, one has a highly specialized corpus for each situation one wishes to learn a script for, but such specialized corpora rarely exist.

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 use complex language, conflate steps and recommendations, and often involve complex 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. Because *The Restaurant Game* is an existing virtual game, Orkin and Roy have an *a priori* known set of actions that can occur in restaurants (e.g., sit down, order, etc.). Boujarwah et al. (2011) and Boujarwah, Abowd, and Arriaga (2012) propose an approach in which workers use natural language to describe event sequences for a given situation such as going to a restaurant. Building off the restaurant game, they learn a probabilistic model of event transitions starting from natural language instead of known event types. While promising, their approach is slow and costly due to multiple rounds of crowdsourcing and does not produce a generalized model. Our work further builds off their work by represent knowledge in a more generalized fashion—as scripts—and attempting to automate as much of the script learning process as possible.

Crowdsourcing Narrative Examples

To learn a script for a particular, given situation we use the following three-step process. First, we query crowd workers to provide linear, natural language narratives of the given situation. Second, we identify the events—the primitive activities that comprise the script. We identify sentences in the crowdsourced narratives that describe the same thing and enumerate these common steps as the events. Third, we construct the script. These steps are described in subsequent sections of this paper.

For our purposes, a script, regardless of situation, 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 capture causal information, which are important for narrative comprehension (Graesser, Singer, and Trabasso 1994) and facilitate many story understanding tasks. A set of *before* relations allows for partial orderings, which can allow for variations in legal event sequences for the situation.

Our approach starts with a query to a Crowdsourcing Web service such as Amazon Mechanical Turk, requesting people to write short narratives about a particular situation. After some amount of time, a small, highly specialized corpus of examples of how the task can be carried out is acquired. To facilitate the learning of events and the probabilistic before relations, our system includes precise instructions to the anonymous workers that make the script-learning task easier. First, we ask that 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 patron, 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.

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. Fragments of crowdsourced narratives.

Second, we ask workers to segment the narrative into events such that each sentence contains a single activity. We refer to each segmented activity as a *step*. Third, we ask workers to use simple natural language such as using one verb per sentence. Figure 1 shows two narratives about the same situation.

Event Learning

Event learning is a process of determining the primitive units of action to be included in the script. Unlike Orkin and Roy (2009) where the set of possible actions are known in advance, we must learn the events from natural language descriptions. We must overcome several challenges. First, there may be different ways to perform a task and therefore narratives may have different steps, or the same steps but in different order. Second, different workers may have used different words to describe the same steps. Third, crowdsourced narratives may omit steps. By working from natural language descriptions of situations, we learn the salient concepts used by a society to represent and reason about common situations.

Our approach is to automatically cluster steps from the narratives based on similarity between sentences such that clusters come to represent the consensus events that should be part of the script. There are many possible ways to perform clustering; below we present the technique that leverages the simplified language use encouraged by our crowdsourcing technique. The semantic similarity between steps from different narratives is computed with the main verb, the main actor, and the verb patient if any. Based on this similarity, steps are clustered in order to identify the core set of events.

Semantic Similarity Computation

Based on the results of the Stanford parser (Klein and Manning 2003), we identify the actor, verb, and the most salient non-actor noun using a rule-based approach. For each verb or non-actor noun, we perform word-sense disambiguation to identify the best WordNet synset (Pedersen and Kolhatkar 2009). After that, we use the WordNet Gloss Vector technique (Patwardhan and Pedersen 2006) to compute the cosine similarity metric for any two weighted term vectors for the desired synsets, which is the semantic similarity [0…1] between two verbs or two nouns.

Event Clustering

We model event learning as an unsupervised clustering of steps, making use of the semantic information pre-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. While k-Means cluster is the more common unsupervised clustering algorithm, we find our combined set of similarity measures does not readily allow for a mean value to be computed for a set of narrative steps. Clustering is performed based on similarity between steps, computed as the weighted sum of the following elements:

- Semantic similarity of verbs
- Semantic similarity of nouns.
- The difference in event location—we compute a step's location as the percentage of the way through a narrative.

Event location helps disambiguate semantically similar steps that happen at different times, especially when a situation is highly linear with little variation. Note that 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.

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

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. Our experience suggests that on Mechanical Turk, each story can be acquired at the cost of $0.40 to $1.00. 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

Table 1. Crowdsourced data sets.

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

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**

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 actual 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 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., *John attempts to kiss Sally*) that appear rarely. The 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.

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 can be 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. 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. Chambers and Jurafsky train their model on the Timebank corpus (Pustejovsky et al. 2003), which uses temporal signal words. Because we are able to leverage a highly specialized corpus of narrative examples of the desired situation, we can probabilistically determine ordering relations between events directly from the crowd-sourced narrative examples. The result of this process is a script-like structure called 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.

Initial Script Construction

Script construction is the process of identifying the plot graph that most compactly and accurately explains the set of crowdsourced narrative examples. 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 event representing event e_2. If s_1 and s_2 appear in the same input narrative, and s_1 appears before s_2, 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 of a particular hypothesis h is $p_h = k/n$ where n is the number of observations and k is the observations that support h. We also measure the confidence of each hypothesis (cf. Wang 2009); a small number of observations for a hypothesis will result in low confidence. We cannot assume prior distributions of orderings between arbitrary events, so we use the imprecise Dirichlet model (Walley 1996) to represent the uncertainty of opposing relations as the interval between the most pessimistic and the most optimistic estimates. The upper and lower estimates of the probability are $p_h^+ = (k+s)/(n+s)$ and $p_h^- = k/(n+s)$, respectively, where the parameter s can be considered as a number of observations whose outcomes are hidden. Our confidence in a

probability is $c_h = 1 - \left(p_h^+ - p_h^-\right) = 1 - s/(n+s)$.

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 which 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 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 $D_N(e_1, e_2)$. $D_N(e_1, e_2)$ is the normative distance from e_1 to e_2 averaged over the entire set of narratives, computed as the average of the distance between two steps s_1 and s_2 in a narrative that belong to e_1 and e_2, respectively. $D_G(e_1, e_2)$ is the distance of e_1 and e_2 on the graph, which is also the minimum number of events that must occur between e_1 and e_2 in all totally ordered sequences consistent with the *before* relations of the plot graph. 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 will 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 2). 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 an 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 3 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.

$\boldsymbol{Q}$:= all of events (e_1, e_2) where e_2 is reachable from e_1 or unordered
Foreach $(e_1, e_2) \in \boldsymbol{Q}$ in order of decreasing $D_N(e_1, e_2) - D_G(e_1, e_2)$ **do:**
  $\boldsymbol{E}$:= all events such that for each $e_i \in \boldsymbol{E}$, $D_G(e_1, e_i) = D_N(e_1, e_2) - 1$
  Foreach $e_i \in \boldsymbol{E}$ **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 $e_i \rightarrow e_2$ now 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 2. The plot graph improvement algorithm.

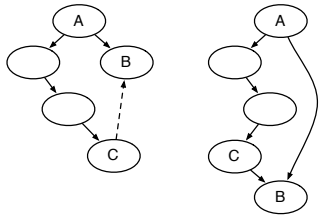

Figure 3. Compensation for errors between pairs of events.

We find this approach to be effective at reducing graph error when T_p relatively high (> 0.5) and $T_c \approx 0.4$. A conservative T_p initially discard 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, making the graph more linear.

Results and Discussion

Figure 4 shows the plot graph learned for the fast food restaurant situation. This plot graph was 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. Some edges are omitted from the figure that do not affect the partial ordering. The asterisks in Figure 4 indicate edges that were added during graph improvement.

The performance of the graph improvement algorithm is summarized in Table 3, which were averaged across 128 different paremeter configurations. 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.

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

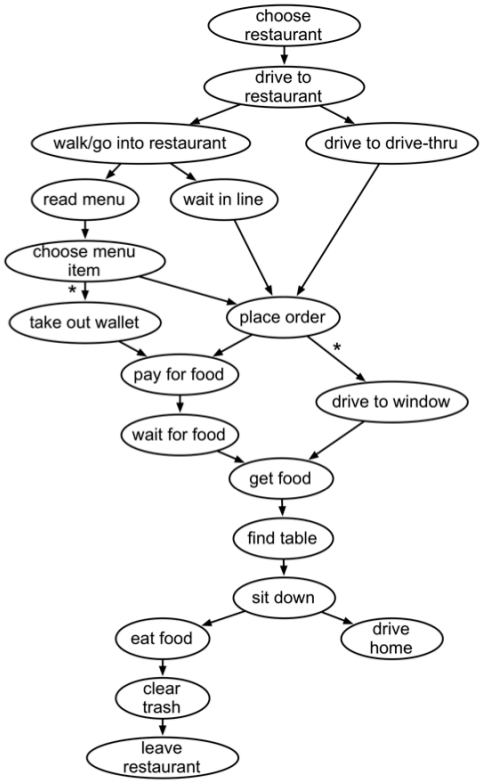

Figure 4. A plot graph generated for the restaurant situation.

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 theoretically possible to eliminate the cycle by splitting the event cluster in the cycle with the highest inter-cluster variance. We have not yet implemented this procedure, however. 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 a situation accepted by the script.

Closely inspecting Figure 4, we note that *before* relations do not always imply 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. Either event is sufficient for *placing an order*. They both appeared in the graph because there are two main variations to the situation, walk-in and drive-through (this also accounts for the unordered *leave restaurant* and *drive home* events). On the other hand, *reading the menu* and *waiting in line* are both necessary for *placing an ordering*. To make coherent decisions when multiple paths through the graph exist, it is necessary to differentiate between causal necessity, causal sufficiency, and simple temporal precedence in future work. As crowd workers habitually omit events that are considered too obvious, we found it difficult to apply traditional probabilistic definitions of causality on the crowdsourced narratives. A promising direction is to ask crowd workers to provide causal information by answering questions about causal counterfactuals. Counterfactuals have been a valuable means of determining causal necessity and sufficiency and we propose to adapt the techniques of Trabasso and Sperry (1985). We expect to be able to minimize the number of questions asked to crowd workers by exploiting the temporal structures we already learned.

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%

Conclusions

Human computation and crowdsourcing provides direct access to humans and the ways they express experiential knowledge. Guided by our five requirements for acquiring sociocultural knowledge, we demonstrated that we could obtain reasonable scripts of common sociocultural situations. Cost-effectiveness is due to automated aggregation of worker effort. Natural interactions are achieved by allowing humans to express their knowledge intuitively as narrative. Our technique is tolerant of situational variations and can accommodate omitted events that are natural consequences of human elicitation of commonsense knowledge. Proactivity is left for future work, although we have identified how human computation can be advantageous. The potential contributions of our preliminary work are (a) the use of stories as a means for knowledge transfer from human to computer via a specialized corpus, (b) the use of explicit instructions to crowd workers to control for natural language use, and (c) a procedure for compiling script-like knowledge structures from story examples. We believe that this is a first step toward rapidly and automatically acquiring functional sociocultural knowledge through the use of anonymous human storytellers. Our approach has the potential to significantly alleviate the knowledge-authoring bottleneck that has limited many practical intelligent systems.

Acknowledgements

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

References

Bean, D. and Riloff, E. 2004. Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. *Proc. of 2004 HLT/NAACL Conference*.

Boujarwah, F., Abowd, G., and Arriaga, R. 2012. Socially Computed Scripts to Support Social Problem Solving Skills. *Proc. of the 2012 Conf on Human Factors in Computing Systems*.

Boujarwah, F., Kim, J.G., Abowd, G., and Arriaga,R. (2011). Developing Scripts to Teach Social Skills: Can the Crowd Assist the Author? In *Proceedings of the AAAI 2011 Workshop on Human Computation*.

Brody, S. 2007. Clustering Clauses for High-level Relation Detection: An Information-theoretic Approach. *Proc. 43rd Annual Meeting of the Association for Computational Linguistics*.

Chambers, N., and Jurafsky, D. 2009. Unsupervised Learning of Narrative Event Chains. *Proceedings of ACL/HLT 2009*.

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.S., Bejan, C.A., and Sagae, K. 2011. Commonsense causal reasoning using millions of personal stories. *Proc. of the 25th Conference on Artificial Intelligence*.

Graesser, A., Singer, M., and Trabasso, T. 1994. Constructing Inferences During Narrative Text Comprehension. *Psychological Review*, 101: 371-395.

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., and Oates, T. 2010. Mining Script-like Structures from the Web. *Proc. of the NAACL/HLT 2010 Workshop on Formailism and Methodology for Learning by Reading*.

Klein, Dan and Manning, Christopher D. 2003. Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

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*.

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

Orkin J. and Roy, D. (2009). Automatic Learning and Generation of Social Behavior from Collective Human Gameplay. *Proc. of the 8th International Conference on Autonomous Agents and Multiagent Systems*.

Patwardhan, S. and Pedersen, T. 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. *Proc. of the EACL Workshop on Making Sense of Sense*.

Pedersen, T. and Kolhatkar, V. 2009. WordNet::SenseRelate:: AllWords - A Broad Coverage Word Sense Tagger that Maximimizes Semantic Relatedness. *Proc. of the ACL 2009 Conference*.

Pustejovsky, J., Hanks, P. Saurí, R., See, A., Gaizauskas, R., Setzer, A. Radev, D. Sundheim, B., Day, D. Ferro, L. and Lazo, M. 2003. The TIMEBANK Corpus. *Proc. 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.*

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

Singh, P., and Williams, W. 2003. LifeNet: A Propositional Model of Ordinary Human Activity. *Proc. of the 2nd International Conference on Knowledge Capture*.

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

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):3-57.

Wang, P. 2009. Formalization of Evidence: A Comparative Study. *Journal of Artificial General Intelligence* 1:25-53.

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