Gathering Background Knowledge for Story Understanding through Crowdsourcing

Christos T. Rodosthenous[1] and Loizos Michael[2]

1 **Open University of Cyprus**
Nicosia, Cyprus
`christos.rodosthenous@ouc.ac.cy`

2 **Open University of Cyprus**
Nicosia, Cyprus
`loizos@ouc.ac.cy`

Abstract

Successfully comprehending stories involves integration of the story information with the reader's own background knowledge. A prerequisite, then, of building automated story understanding systems is the availability of such background knowledge. We take the approach that knowledge appropriate for story understanding can be gathered by sourcing the task to the crowd. Our methodology centers on breaking this task into a sequence of more specific tasks, so that human participants not only identify relevant knowledge, but also convert it into a machine-readable form, generalize it, and evaluate its appropriateness. These individual tasks are presented to human participants as missions in an online game, offering them, in this manner, an incentive for their participation. We report on an initial deployment of the game, and discuss our ongoing work for integrating the knowledge gathering task into a full-fledged story understanding engine.

1998 ACM Subject Classification I.2.4 Knowledge Representation Formalisms and Methods

Keywords and phrases story understanding, knowledge representation, crowdsourcing, reasoning

Digital Object Identifier 10.4230/OASIcs.CMN.2014.154

1 Introduction

A defining characteristic of human intelligence is our ability to comprehend stories based on previous experiences and acquired knowledge. These experiences and beliefs act as background knowledge for the comprehension task. To create a system able to understand stories, we must first devise a method for gathering such background knowledge from some appropriate source in a form that can be later used by a story understanding engine.

This paper describes our ongoing work for this knowledge acquisition task. It focuses on describing a method for acquiring background knowledge through crowdsourcing, and it initiates an investigation of whether a *fully* crowdsourced method for knowledge acquisition is feasible, and competitive against other automated or semi-automated approaches.

The paper is organized as follows: First, the formal framework used to represent and reason with the background knowledge is analyzed, and our approach is compared to other existing works. The methodology used to gather background knowledge is then presented, as a sequence of steps needed to get from raw text to structured knowledge. We cast our methodology as a crowdsourcing task, and demonstrate how Games With A Purpose (GWAPs) can be used to implement it. Finally, an empirical setting and results from an initial deployment of our developed GWAP are presented. We conclude with future work.

© Christos T. Rodosthenous and Loizos Michael;
licensed under Creative Commons License CC-BY
5th Workshop on Computational Models of Narrative (CMN'14).
Editors: Mark A. Finlayson, Jan Christoph Meister, and Emile G. Bruneau; pp. 154–163
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Related Work and Background

One can think of a story understanding engine as comprising three main modules:

- a module for converting stories from a given modality (e.g., text) to a formal representation
- a module for gathering background knowledge and representing it formally
- a module for reasoning by integrating story information with background knowledge

We discuss related work in terms of these three modules. Since the early seventies, a plethora of systems and models have been developed for story understanding. Charniak [4] proposed a story comprehension model for answering questions about children stories by relating stories to real-world background knowledge. The same author also proposed the Ms. Malaprop [5] system, which answers questions about simple stories dealing with painting. That system uses stories entered in a semantic representation format and answers questions using rules of the same format. Hirschman et al. [13] described the work done on Deep Read, an automated reading comprehension system that accepts stories and answers questions about them. Mueller [26] proposed a system for modelling space and time in narratives about restaurants, which involved the development of an information extraction tool to convert narrative texts into templates about the dining episodes discussed in the narratives. These templates were used for constructing commonsense reasoning problems.

Most of these systems are focused on a specific domain or subject area, like terrorism, painting, dinning in restaurants, etc., and require specific background knowledge based on the respective topic. The story comprehension level of the majority of these systems is also limited to the basic events covered in each story and the key actors involved.

In an analogous context, Gordon and Schubert [11] proposed a method for acquiring conditional knowledge by exploiting presuppositional discourse patterns to create general rules. Clark and Harrison [6] developed a system able to extract simple statements of world knowledge from text, which aims to improve parsing and the plausibility assessment of paraphrase rules used in textual entailment.

The importance of background knowledge in story comprehension is also backed up by reports coming from Psychology; see, for example, the work by Diakidoy et al. [7, 8] and references therein. Certain researchers in the field claim that the appropriate knowledge for this type of systems is based on general axiomatic formulations of different facets of the commonsense world [12]; others claim that symbolic representations [29] or concrete rules [14] are the right way to represent background knowledge, and yet others claim that routine behavioral activity that operates using purely procedural representations is the appropriate format [1]. A hybrid approach is proposed in [27] and [31], where background knowledge is allowed to be represented in a variety of formats. This diversity makes it more likely to gather appropriate background knowledge for whatever commonsense problem one is faced with at the moment. We adopt the approach that background knowledge is represented in terms of rules (cf. Section 2.1), which correspond to loose associations between concepts, in line with relevant psychological evidence [17, 21].

According to Mueller [25], a story understanding engine should be able to acquire broad and deep background knowledge. There are many initiatives for collecting and distributing background knowledge, including Open Mind [31], ConceptNet [20], WordNet [9], PropBank [16], FrameNet [2], etc. Most of these initiatives acquire knowledge by posing questions to volunteers on specific subjects, and represent knowledge in an unstructured (textual) or semi-structured (network of keywords and relations) form.

During the last several years, we have witnessed the blossom of crowdsourcing techniques and more specifically Games With A Purpose. Crowdsourcing is a relatively new term and is typically defined as 'a strategy that combines the effort of the public to solve a problem or produce a resource.' [36]. GWAP [34] is a genre of crowdsourcing and is best described by existing applications such as the ESP game [35] and Verbosity [33]. We adopt, in this work, the use of GWAPs as the mechanism for acquiring commonsense knowledge.

There are certain other attempts that use GWAPs to acquire commonsense knowledge, such as the Common Consensus game [19], which aims to collect commonsense knowledge from people's everyday goals, and the Restaurant Game [28], where player actions and behavior in a virtual restaurant world are recorded, encoded, and visualized on a plan network. Boyang et al. [18] proposed a system for creating narratives through crowdsourcing by using the representation of plot graphs.

2.1 Knowledge Representation and Reasoning

We are interested in obtaining background knowledge that can be used in the context of multiple stories. We take the approach of representing knowledge in a structured form, using a high-level version [22] of the Event Calculus [30], with the aim of exploiting formal reasoning systems (e.g., [7, 8]). In the sequel we use the following terminology and notation:

A fluent F is an object whose value can change through the course of time like quantities or propositions [30]. An action A is an event that occurs at a specific time-point. A literal L can be a fluent or an action, or their negation. The following types of rules are used:

- ϕ `implies` L: Denotes a formula ϕ over actions and fluents that implies literal L. Rules of this type correspond to constraints that hold at each story time-point. The rule person(X) `implies` can(X,think), for example, intuitively means that every person X can think.
- ϕ `causes` L: Denotes a formula ϕ over actions and fluents that causes literal L. Rules of this type capture the conditions ϕ whose presence at some time-point is sufficient to change the state of L at the next time-point. The rule attack(X,Y) `causes` war(X,Y), for example, intuitively means that when X attacks Y it causes a war between them.

A story is taken to be a sequence of literals that hold or occur at certain time-points [23].

3 Gathering Background Knowledge

Following our main goal of investigating whether a fully crowdsourced approach suffices for knowledge acquisition, we propose a general scheme for going from raw text to background knowledge represented in terms of structured rules. We illustrate the steps of our methodology below, using the following simple story snippet as a running example:

Story snippet: A cat chased the mice. The mice managed to hide in a nearby hole.

Step 1. A story is selected and is split into sentences, using punctuation marks to determine the end of each sentence. A sentence is then selected for processing. Human participants are asked to remove articles (e.g., 'a', 'the'), change the tense of verbs (e.g., 'chased' to 'chase') and lemmatize words (e.g., 'mice' to 'mouse'). This step converts sentences and words to a simpler form by reducing inflectional forms, and removing stop words.

Selected sentence: A cat chased the mice.
After processing: cat chase mouse

Step 2. Human participants are asked to identify nouns and verbs given the previously processed phrases. The outcome will be later used to produce formal expressions, which allow verbs being used as predicate name and nouns being used as predicate arguments.

Selected phrase: cat chase mouse
After separation: {cat, mouse} are nouns, and {chase} is a verb

Step 3. Predicates are constructed using verbs and nouns from the previous step. More specifically, human participants choose which verbs to use as predicate names and which nouns to use as predicate arguments. In addition to nouns, each constructed predicate can be used as an argument for new predicates that are created, leading to higher-order predicates. Human participants are required to choose whether a predicate is an action or a fluent.

Selected words: {cat, mouse} are nouns, and {chase} is a verb
Formal expression: chase(cat,mouse) is an action

Step 4. The next step seeks to identify logical rules that are built on the identified predicates. What is expected here is for the human participants to introduce new predicates that are not explicitly present in a sentence, but are implied by it, and relate those new predicates to the existing ones in the form of rules. For each rule, human participants are asked to specify whether this rule causes or implies the deduced predicate.

Selected predicate: chase(cat,mouse)
Possible rule 1: chase(cat,mouse) `causes` fear(mouse,cat)
Possible rule 2: chase(cat,mouse) `implies` can(cat,run)

Step 5. In the penultimate step, human participants generalize previously identified rules. For each rule certain predicates and arguments can be chosen and replaced with variables. When an argument α is replaced with a variable V, a new predicate of the form $\alpha(V)$ is appended to the body of the rule. Human participants can choose whether this predicate should be retained. Effectively, this step transforms each rule to a form that is applicable more generally and not only in the context of the story or sentence from which it originated.

Selected rule: chase(cat,mouse) `implies` can(cat,run)
Possible generalized rule 1: cat(X) `and` chase(X,mouse) `implies` can(X,run)
Possible generalized rule 2: chase(X,Y) `implies` can(X,run)

Step 6. During the final step the acquired knowledge is validated. First, a sentence other than the one from which a given rule originated, is selected. Human participants are asked to verify whether the conditions in the body of the rule are met in the context of the selected sentence. If they are, human participants are asked to decide whether the head of the rule follows from the sentence. If the player answers affirmatively to the first question then the rule receives a positive applicability vote; otherwise, the rule receives a negative applicability vote. If the player answers affirmatively to the second question then the rule receives a positive validation vote; otherwise, the rule receives a negative validation vote.

Selected context: A policeman was chasing a burglar near the town center.
Selected rule: chase(X,Y) `implies` can(X,run)
Results: The conditions in the body of the rule are met in the context of the selected sentence, and the head of the rule follows from the selected sentence. Thus, the rule receives a positive applicability vote and a positive validation vote.

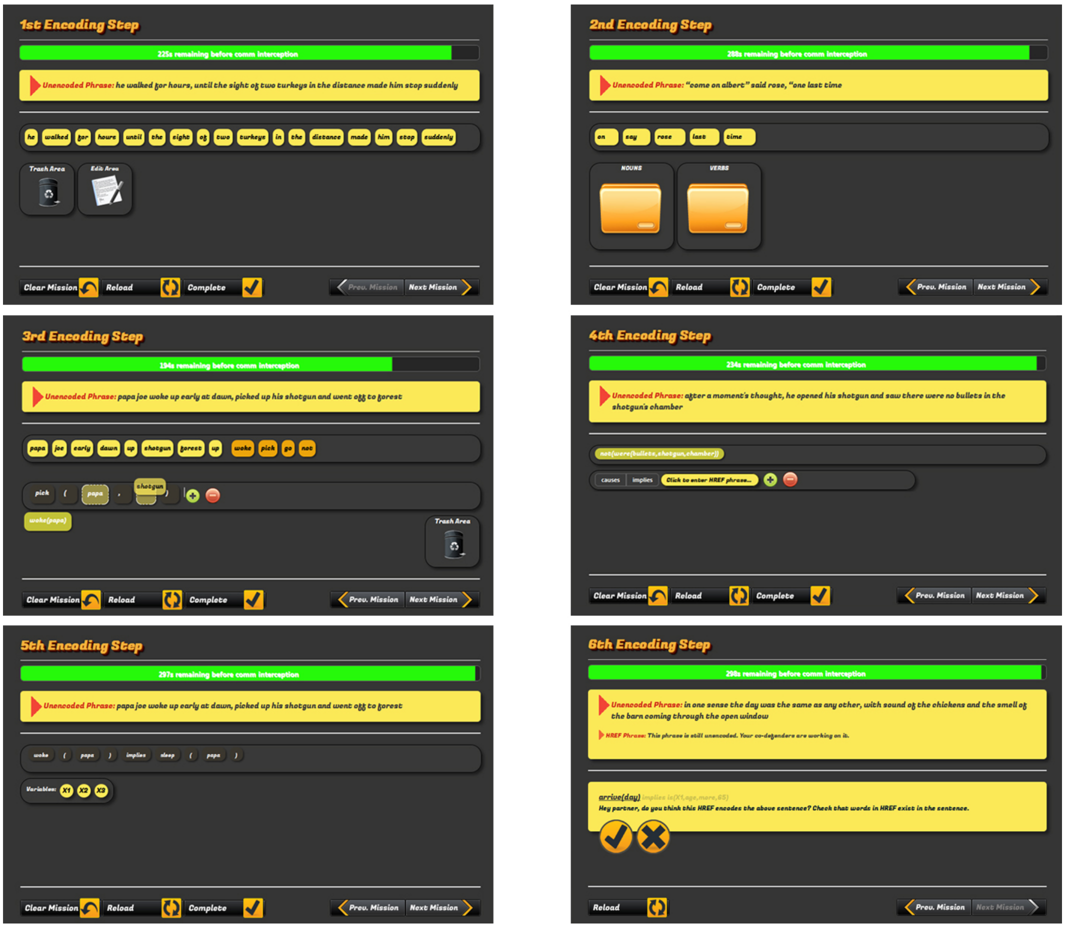

Figure 1 Screenshots of the six game missions in the 'Knowledge Coder' game.

After all six steps are completed, the resulting background knowledge comprises those rules that have been found to be sufficiently applicable and sufficiently validated.

4 Crowdsourcing through a GWAP

The proposed methodology implicitly assumes that human participants are knowledgable, honest and willing to participate. Since these assumptions might not necessarily hold in practice, certain measures need to be taken to counter the possibly negative effects of the actions of less knowledgable or honest participants. One such measure is already present in the methodology. The multiple steps it comprises reduce the possibility of user error and the complexity for novice human participants, allow for easier control of the outcomes of each step, and facilitate the integration with knowledge understanding systems.

We adopt the use of GWAPs for the crowdsourcing of knowledge acquisition as a way of motivating people to participate [34]. Our developed game is called 'Knowledge Coder' (see Figure 1) and a prototype version is accessible online at: `http://cognition.ouc.ac.cy/narrative`.

Our approach falls into the output-agreement games template [34], requiring players to agree on the same output they produce. The game follows closely the methodology described in the previous section, with each step corresponding to a 'mission' in the game.

The game story takes place in the near future, where planet Earth is captured by alien forces capable of intercepting human communications in natural language. Players are asked to join the resistance forces and help their co-defenders encode human knowledge in a structured form that is not readable by aliens, and thus guard it from being intercepted.

Players are introduced to a game environment containing a mission instructions area, a time countdown bar, a high scores area, and an active mission area. Players also have access to mission specific instructions and online help during game play.

As with other games, players are encouraged to play using competitive motives [10]. For each successful mission attempt, players are rewarded with points that are added to their total score. Players are also rewarded with extra points when other players contribute and verify the former players' mission results and vice versa. These extra points are used to separate the knowledgeable and honest players from the rest. After a player reaches a certain score, an award is issued and added to the player's profile. These methods are commonly applied techniques to encourage and promote competition among players in games [15].

A common problem in online games is cheating through, for instance, communication between players outside the game [24]. To reduce such effects, missions are time-bounded to prevent players from using external help to complete them. The anonymity of players is pursued and no contact details are made available throughout the game play. Also, each player's Internet address is recorded and associated with each attempt on a mission, so that individual players masquerading as two or more different players are detected and are filtered out. Finally, every mission is initiated with a random sentence, so that the probability of two players attempting to work on the same instance of a task is minimized.

Players can provide feedback through the game interface. Feedback submitted is valuable both for debugging purposes and for further game development. Players can request new features, changes to the user interface, or extra missions, or suggest improvements.

5 Empirical Setting and Results

For our initial empirical evaluation of the game we prepared an evaluation process using a small group of people and two stories loaded into the game. Both chosen stories were short and used simple English words. For the purposes of this evaluation we selected two Aesop Fables: 'The Oxen and the Butchers' and 'The Doe and the Lion' [32].

Five participants were trained on how to play the game on a test deployment of the game. This group included both men and women aged eighteen and above, all with a high school education, and with some of them enrolled in a university. All missions were presented and each player had the opportunity to familiarize themselves with the look and feel of the game. For the purposes of the experiment, each player created a game account. The game was available for one week, at the end of which each player was asked to complete a questionnaire. All knowledge gathered was analyzed, and our conclusions are presented below.

5.1 Analysis of Results

We collected approximately one hundred user-generated rules; Table 1 presents some relevant information. Below we present and discuss a sample of the collected rules.

```
R1: horn(X) and assemble(X) and carry(purpose) and sharpen(X) and
    assemble(certain,X,carry(purpose)) implies have(ox,horns)
```

```
R2: assemble(day) and carry(purpose) and sharpen(horn) and
    assemble(certain,day,carry(purpose)) implies prepare(ox,war)
```

```
R3: beast(X) and throw(Y,mouth,X) implies kill(X,Y)
```

```
R4: beast(X) and man(Y) and doe(Z) and exclaime(Z) and
    escape(Z,Y) and throw(Z,X) implies kill(X,Z)
```

As one can observe, rules R1 and R2 are too specific and tightly coupled to the story used to generate them ('The Oxen and the Butchers'). This level of specificity is inappropriate for gathering broad background knowledge. The metric of applicability can be used to filter such rules out. By requiring rules with high applicability, we are more likely to end up with rules like rule R3 which can be usefully applied in almost any story with wild animals. The fact that the majority of the rules produced by the first five steps of our methodology did not receive a high applicability score during the sixth step, suggests the need for an additional incentive in the game so that players produce simpler and more general rules. Such an incentive, for example, would allow players to suggest the deletion of predicates man(Y), doe(Z), exclaime(Z) and escape(Z,Y), from rule R4 to produce a rule similar to rule R3.

Note that rule R4 includes a misspelled predicate name (i.e., 'exclaime' instead of 'exclaim'), demonstrating that output-agreement does not guarantee that the gathered knowledge is error-free, and that additional incentives might be needed to reduce such errors.

5.2 Player Feedback

After completing the game, each player was asked to complete a questionnaire for assessing the game design, concept, usability, enjoyment and other factors such as playing time, game scoring, etc. Feedback was also requested on how well players understood the instructions given for each mission and the time needed for them to comprehend them before starting playing. Finally, players were asked whether missions are relevant to the game concept and what they would like to see changed for the game to become more engaging.

By analyzing this feedback we conclude that players found the game story interesting and that they would be willing to advertise the game to their friends. Most players found the first two missions (i.e., 'sentence processing' and 'verb and noun identification') easy to play and the instructions given informative. For the next two missions (i.e., 'predicate construction' and 'rule construction'), players seemed to require some time before understanding fully what they were expected to do. These two missions were also characterized as the most interesting ones and kept players engaged throughout the game play.

Four out of five responders characterized the fifth mission (i.e., 'rule generalization') as not very challenging, since they understood that they only had to replace arguments with variables. On the one hand, this feedback suggests a misunderstanding on the part of the players on what they were expected to do, which can be avoided by improving the mission instructions. On the other hand, this feedback is in line with the acquisition of not highly applicable rules, which suggests the need for stronger incentives to simplify the rules.

Several of the comments received concerned the creation of a tablet and mobile version of the game and integration with social media for posting score to the players' friends and

Table 1 Relevant information from the experimental deployment of the 'Knowledge Coder' game.

Number of stories	2
Number of sentences	7
Number of players	5

Number of rules generated	93
Number of causality rules	15
Number of implication rules	78

contacts. One responder suggested that more languages should be available for the game.

6 Conclusion and Future Work

Designing an engine that can handle broad background knowledge for story understanding is far from being a trivial task, due to the fact that this knowledge is not given explicitly in the actual story text. In this paper we have presented our initial work on developing a crowdsourced solution to the problem of acquiring such knowledge directly from humans. The background knowledge gathered from our developed game offers some initial encouraging results in terms of the feasibility of our methodology. With improved instructions and incentives we expect to address the problem with the acquisition of highly applicable rules.

We could also explore different paths in the methodology used. Instead of using the Event Calculus, we could consider using the Situation Calculus [3] for representing the acquired knowledge without the need to reference particular time-points. An important enhancement to our methodology would be the addition of an extra step to denote preferences among pairs of rules with conflicting heads. This should also be reflected in the game in the form of an extra mission, after the currently last mission of 'rule evaluation'.

As part of our ongoing work we are implementing these and other improvements suggested by the user feedback that we have received, and plan to deploy the 'Knowledge Coder' game to gather background knowledge and conduct further experiments with more stories and players. A reasoning module will be integrated with the knowledge acquisition module to reason with the acquired knowledge on new unseen stories, offering a means to evaluate the acquired knowledge on the task of interest itself: understanding stories. In particular, we plan to integrate the framework of Diakidoy et al. [7, 8], which has been developed based on psychologically-validated models of narrative comprehension, and whose formal representation of stories and background knowledge closely matches the representation used in our work. Finally, we plan to compare our fully-crowdsourced solution against automated or semi-automated ones for acquiring background knowledge.

Although our work has centered on the task of knowledge acquisition for story understanding, we believe that our methodology is applicable more generally, and can find use in other lines of research that assume as given commonsense knowledge in a structured form.

References

1 Philip E. Agre and David Chapman. Pengi: An Implementation of a Theory of Activity. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI'87)*, pages 286–272, Seattle, Washington, USA, 1987.

2 Sue Atkins, Michael Rundell, and Hiroaki Sato. The Contribution of FrameNet to Practical Lexicography. *International Journal of Lexicography*, 16.3:333–357, 2003.

3 Kristof Van Belleghem, Marc Denecker, and Danny De Schreye. On the Relation Between Situation Calculus and Event Calculus. *The Journal of Logic Programming*, 31(1-3):3–37, 1997.

4 Eugene Charniak. Toward a Model of Children's Story Comprehension. Technical Report AITR-266, Massachusetts Institute of Technology, Cambridge, MA, USA, 1972.

5 Eugene Charniak. *Ms. Malaprop, a Language Comprehension Program.* Institut pour les Etudes Sémantiques et Cognitives Genève: Working papers. Fond. Dalle Molle, 1976.

6 Peter Clark and Phil Harrison. Large-scale Extraction and Use of Knowledge from Text. In *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP'09)*, pages 153–160, New York, NY, USA, 2009.

7 Irene-Anna Diakidoy, Antonis Kakas, Loizos Michael, and Rob Miller. Narrative Text Comprehension: From Psychology to AI. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*, Ayia Napa, Cyprus, 2013.

8 Irene-Anna Diakidoy, Antonis Kakas, Loizos Michael, and Rob Miller. Story Comprehension through Argumentation. In *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA'14)*, Scottish Highlands, UK, 2014.

9 Christiane Fellbaum. WordNet. In Roberto Poli, Michael Healy, and Achilles Kameas, editors, *Theory and Applications of Ontology: Computer Applications*, pages 231–243. Springer Netherlands, 2010.

10 Rosemary Garris, Robert Ahlers, and James E. Driskell. Games, Motivation, and Learning: A Research and Practice Model. *Simulation & Gaming*, 33(4):441–467, 2002.

11 Jonathan Gordon and Lenhart K. Schubert. Discovering Commonsense Entailment Rules Implicit in Sentences. In *Proceedings of the EMNLP 2011 Workshop on Textual Entailment (TextInfer'11)*, Edinburgh, Scotland, UK, 2011.

12 Kenneth Haase. FramerD: Representing Knowledge in the Large. *IBM Systems Journal*, 35(3.4):381–397, 1996.

13 Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. Deep Read: A Reading Comprehension System. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, pages 325–332, Stroudsburg, PA, USA, 1999.

14 Jerry R. Hobbs, Mark E. Stickel, Douglas E. Appelt, and Paul Martin. Interpretation as Abduction. *Artificial Intelligence*, 63(1–2):69–142, 1993.

15 Hamari Juho and Eranti Veikko. Framework for Designing and Evaluating Game Achievements. In *Proceedings of the 5th International Conference of the Digital Research Association (DIGRA'11)*, Utrecht, Netherlands, January 2011.

16 Paul Kingsbury and Martha Palmer. From Treebank to Propbank. In *Proceedings of the 3rd International Conference on Language, Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands, Spain, 2002.

17 Walter Kintsch. The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Review*, 95(2):163, 1988.

18 Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. Story Generation with Crowdsourced Plot Graphs. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI'13)*, Bellevue, Washington, USA, 2013.

19 Henry Lieberman, Dustin A. Smith, and Alea Teeters. Common Consensus: A Web-Based Game for Collecting Commonsense Goals. In *Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*, Honolulu, Hawaii, USA, 2007.

20 Hugo Liu and Push Singh. ConceptNet — A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal*, 22(4):211–226, 2004.

21 Danielle S. McNamara and Joe Magliano. Toward a Comprehensive Model of Comprehension. *Psychology of Learning and Motivation*, 51:297–384, 2009.

22 Loizos Michael. Computability of Narrative. In *Proceedings of the 2nd Symposium on Computational Models of Narrative (CMN'10)*, Arlington, Virginia, USA, 2010.

23 Loizos Michael. Story Understanding... Calculemus. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*, Ayia Napa, Cyprus, 2013.

24 Christian Monch, Gisle Grimen, and Roger Midtstraum. Protecting Online Games Against Cheating. In *Proceedings of the 5th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames'06)*, New York, NY, USA, 2006.

25 Erik T. Mueller. Understanding Script-Based Stories Using Commonsense Reasoning. *Cognitive Systems Research*, 5(4):307–340, 2004.

26 Erik T. Mueller. Modelling Space and Time in Narratives About Restaurants. *Literary and Linguistic Computing*, 22(1):67–84, 2007.
27 Peter Norvig. *A Unified Theory of Inference for Text Understanding*. PhD thesis, EECS Department, University of California, Berkeley, January 1987.
28 Jeff Orkin and Deb Roy. The Restaurant Game: Learning Social Behavior and Language from Thousands of Players Online. *Journal of Game Development*, 3(1):39–60, 2007.
29 Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. Artificial Intelligence Series. Psychology Press, 1977.
30 Murray Shanahan. The Event Calculus Explained. In Michael J. Wooldridge and Manuela Veloso, editors, *Artificial Intelligence Today*, volume 1600 of *Lecture Notes in Computer Science*, pages 409–430. Springer Berlin Heidelberg, 1999.
31 Push Singh. The Public Acquisition of Commonsense Knowledge. In *Proceedings of the AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Palo Alto, California, USA, 2002.
32 Pat Ronson Stewart. *Aesop's Fables*. Dover Children's Thrift Classics. Dover Publications, 1994.
33 Luis von Ahn and Laura Dabbish. Labeling Images with a Computer Game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, pages 319–326, New York, NY, USA, 2004.
34 Luis von Ahn and Laura Dabbish. Designing Games with a Purpose. *Communications of the ACM*, 51(8):58–67, August 2008.
35 Luis von Ahn, Mihir Kedia, and Manuel Blum. Verbosity: A Game for Collecting Common-Sense Facts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'06)*, pages 75–78, New York, NY, USA, 2006.
36 Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. Perspectives on Crowdsourcing Annotations for Natural Language Processing. *Language Resources and Evaluation*, 47(1):9–31, 2013.