Using Common-Sense Knowledge in Generating Stories

Sherie Yu and Ethel Ong

Center for Language Technologies, De La Salle University, Manila, Philippines
sherie_yu@yahoo.com, ethel.ong@delasalle.ph

Abstract. A problem with most story generation systems is the lack of an adequately-sized body of knowledge to generate stories from. This paper presents an approach that focuses on providing a large amount of common-sense knowledge to automatic story generators while keeping extensive manual handcrafting of knowledge to a minimum. It does so by combining manually-created resources with freely-available common-sense knowledge in machine-readable format for the generation of stories.

Keywords: Automatic Story Generation, Storytelling Knowledge, Knowledge Representation, Common-sense Reasoning.

1 Introduction

Story generation systems have been used for endowing computational creativity to computers [1], as a tool for writing [2], and as an educational tool [3]. However, computers seem to have a hard time at generating stories that make sense [1]. This is attributed to the lack of an adequately-sized body of knowledge to generate stories from. Systems such as Mexica [4], Picture Books [5] and SUMO Stories [6] make use of manually built resources that contain domain-dependent information, which make them not work well for unexpected inputs. They do not have the same basic general or common-sense knowledge that humans have to reason about everyday life [9].

This paper presents an approach to providing a large amount of common-sense knowledge to story generation systems by combining manually-created resources with freely-available common-sense knowledge in machine-readable format. Such knowledge is often referred to as storytelling knowledge. The next section identifies the types of knowledge needed by our story generator. This is followed by a description of the architecture that utilized the storytelling knowledge. Preliminary results are provided, ending with a discussion of issues and recommendations for future work.

2 Storytelling Knowledge

Our storytelling knowledge adapts Swartjes' [10] two-layer ontology for representing storytelling knowledge. The upper story world ontology contains existing common-sense knowledge resources. These include ConceptNet [7], a semantic network of common-sense concepts classified into thematic categories such as events, causal and

P. Anthony, M. Ishizuka, and D. Lukose (Eds.): PRICAI 2012, LNAI 7458, pp. 838–843, 2012.

affective, which fit with some of the characteristics inherent in stories; VerbNet [14], a semantic lexicon of verbs classified according to abstract classes with thematic roles and frames that are suitable for use by natural language generation systems, the broader area where story generation falls under; and WordNet [15].

The domain-specific world ontology, on the other hand, models elements that are typical to the target domain, in this case, children's stories. The elements include themes, story characters, and events.

Story themes for children usually center on everyday life experiences (such as *going to camp*) and behavior development (such as *honesty* and *bravery*). Characters are given names to identify them from other characters. The interpersonal relationships that often exist between characters in children's stories such as "*ParentOf*" and "*TeacherOf*" are also modeled. Most critical to the representation of characters are the set of roles and traits that influence their goals and ultimately determine the actions they perform. For instance, a character with the role '*student'* and a trait of "*lazy*" may fail to fulfill the goal to "*have good grade*" by choosing to "*play*" rather than to "*study*". On the other hand, a similar character with a trait of "*dishonest*" rather "*lazy*" may choose to "*cheat*" instead.

Events represent the atomic units of a story. They represent actions and states. When represented in the context of a story, each event is associated with a *timepoint*. States are represented as primitives or predicates that signify a particular meaning as illustrated in Table 1 while actions are represented with their respective VerbNet thematic roles as shown in Table 2.

Table 1. Predicates

Predicate	Meaning
HasPossession (?character, ?object)	character owns object
RelationshipChange (?character1, ?character2, -10)	character2 holds ill feelings towards character1

Table 2. Representation of Events

Event	*steal*	*break*	*scold*
Agent	?character1	?object	?character1
Source	?character2	---	---
Patient	---	---	?character2
Theme	?object	---	---

Themes revolve around the behavioral development of the main character based on a moral or virtue. They are represented as rules in the ontology and are patterned after the classical plot structure for children's stories as presented in [16].

3 System Architecture

The architecture shown in Figure 1 is based on a model of story writing that identifies a balance between the plotting of the characters and the author [17]. The prompt is a user input which specifies the basic elements (characters, objects and optional location) that should be present in the story.

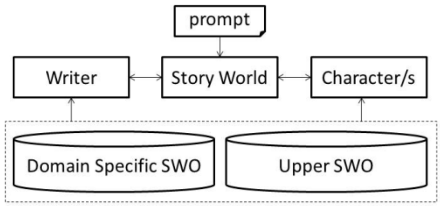

Fig. 1. System Architecture

The writer and the character agents interact through the story world that represents the story being created. It contains a history of the events that have occurred so far with their corresponding *timepoints*. In order to move the story forward, both agents make queries to the storytelling knowledge and store the results to the story world. However, whereas the writer has full access to all the information in the storytelling knowledge, characters have a limited access (i.e., cannot see the themes).

3.1 ConceptNet Disambiguation

Disambiguating between the senses of a concept enables the agent to make better choices. When a '*child*' character agent makes a query for the list of candidate objects a "*child*" could perform with the action "*play*", the upper ontology should return "*toy*" rather than "*piano*" since it is more likely that the sense of "*play*" being referred to is "*being engaged in playful activity*".

Whenever a query is made, the concepts are disambiguated into their proper senses through a modified implementation of the algorithm described by [12]. A *Word Sense Profile* (WSP) or list of terms is constructed for each sense of the ambiguous concept based on WordNet. Then, the relatedness between the terms is measured based on *Normalized Google Distance* (NGD). Lastly, the noisy terms that would decrease the performance of the algorithm are filtered out.

In our story generator, AnalogySpace [13] is used in place of NGD to measure the relatedness between terms. Filtering of noisy terms it not performed. Furthermore, the part-of-speech (POS) tags of each concept in the assertion were identified prior to creating the WSP to optimize the performance of the disambiguation.

Once disambiguated, the concepts are now associated with their WordNet keys to supplement the current ConceptNet labels. Furthermore, verbs get a VerbNet *class id* information since the verbs in VerbNet contain mappings to WordNet.

3.2 Character Goal Formulation

To simulate the plotting of the characters' individual goals and plans, a mechanism for formulating character goals is followed. Every main character agent in the story queries the upper story world ontology for a goal to pursue. For each role assigned to a character, a pool of candidate goals is obtained from ConceptNet through the query:

```
Desires (?role, goal)
```

A random goal for a character is selected after factoring in the scores of each "*Desires*" assertion in the goals pool. For instance, if the query, "*Desires ('child', goal)*" returns the following assertions and their respective scores:

```
Desires (child, learn) +2, Desires (child, play) +5
```

Then "*Desires (child, play)*" would have a better chance of being selected.

3.3 Character Goal Completion

Each character goal can be completed only by the character that formulated the goal. A higher priority is given to actions from the domain-specific ontology.

Querying the domain-specific ontology, events at the previous *timepoint* must satisfy the preconditions of an action whose post-condition matches with the goal. A '*dishonest*' character might achieve the goal '*good grade*' with the action '*cheat*' if it had '*dishonest*' and '*good grade*' as its pre- and post-conditions, respectively.

Querying the upper story world ontology is a bit more complicated. First, AnalogySpace [13] is used to obtain the top *n* similar concepts. A goal is selected from this list of concepts with the highest probability given to the original concept. For each goal, the events that achieve the goal are selected through the following queries:

```
MotivatedByGoal(?event,goal), HasPrerequisite(goal,?event)
Causes(?event,goal), UsedFor(?event,goal)
```

The results are stored disjointly so that any single action can be performed in order to achieve the goal. For the "*good grade*" goal, some of the possible actions that could lead to its achievement are "*attend class, study hard*". These actions are represented in the way actions are described in the domain-specific ontology since they have also been disambiguated and therefore assigned their particular WordNet senses.

3.4 Plot Progression

At every *timepoint*, the writer inspects the story world to determine if the intended story theme has been realized. Since the themes concern behavior development from a negative trait to a positive one (e.g. a "*dishonest character*" learning to be "*honest*"), the rules are generalized as "*when a main character performs an action based on a negative trait that leads to a negative consequence, the character realizes his/her mistake and does the right thing in the future*". Performing an action based on a trait is already embedded into the mechanism by which a character agent chooses an action. The link between the negative action and the consequence that follows is provided by profluence. The creation of a similar circumstance for allowing the character to do the right thing in the future is the task of the writer.

Table 3. Queries for Consequential Progression

Query	Meaning
Causes(current,?event)	What does the current event cause to happen?
HasSubevent (current,?event)	What normally happens when current occurs?

3.5 Story Profluence

Profluence is the logical progression of events. This is manifested through the completion of character goals; and the consequential and motivational progressions as shown in Tables 3 and 4 respectively. The current goal is labeled as *current*.

Table 4. Queries for Motivational Progression

Query	Meaning
MotivatedByGoal (current, ?event)	What events motivate current to happen?
CausesDesire (?event, current)	What events can possibly motivate current? (special case: current must be a goal)

4 Preliminary Results

ConceptNet 4 is used in this system. Only assertions with scores of at least two were considered. The number of concepts, *n*, has been limited to 1.

Listing 1 shows a sample trace of character goals, story world state and character actions. The theme selected is for that of a *greedy* character. The prompt specified only the characters that should be included in the story: *Danny who is greedy* and *Hannah who is stingy*. The two plot points that completed the plot for "*greedy*" were "*steal*" and "*punch*" in which *Hannah punched Danny for stealing her candy*.

Listing 1. Sample Output

```
Timepoint 0:  goal:HasPossession(Danny,candy)
              state:HasPossession(Hannah,candy)
              state:!HasPossession(Danny,candy,)
Timepoint 1:  action:steal(Danny,Hannah,candy,Danny)
Timepoint 2:  state:HasPossession(Danny,candy)
              state:RelationshipChange(Hannah,Danny,-10)
              state:EmotionalState(Danny,happy)
Timepoint 3:  state:EmotionalState(Hannah,anger,Danny)
              action:dance(Danny)
Timepoint 4:  action:punch(Hannah,Danny)
```

5 Conclusion

The paper presented a preliminary work on providing a story generator with existing common-sense knowledge. It describes the representation of the storytelling knowledge as well as how the story generator would make use of this knowledge.

An evaluation system must be devised in close communication with an evaluator (possibly a story writer) to identify an evaluation scheme and the most appropriate story representation to use. The representation must also contain the appropriate contextual information to allow for conversion into natural language text in the future.

ConceptNet was built on user-supplied knowledge through the Open Mind Common Sense (OMCS) project [7] in which common-sense was acquired from the general public online. Despite obtaining a high evaluation score regarding the accuracy of information contained in this resource, the generated output may be inappropriate for children and may also be inaccurate. Some form of automatic validation of an assertion or concept before usage into the system would be beneficial to address the inaccuracy concern.

References

1. Lonneker, B., Meister, J.C.: Dream On: Designing the Ideal Story Generator Algorithm. In: ACH and the ALLC, pp. 1–3. University of Victoria, Victoria (2005)
2. Riedl, M.O.: Narrative Generation: Balancing Plot and Character. Ph.D. Dissertation. North Carolina State University (2004)
3. Rowe, J.P., Mcquiggan, S.W., Lester, J.C.: Narrative Presence in Intelligent Learning Environments. In: AAAI Symposium on Intelligent Narrative Technologies, Arlington, Virginia, pp. 126–133 (2007)
4. Péréz y Peréz, R., Sharples, M.: MEXICA: A Computer Model of a Cognitive Account of Creative Writing. JETAI 13(2), 119–139 (2001)
5. Ong, E.: A Commonsense Knowledge Base for Generating Children's Stories. In: AAAI 2010 Fall Symposium Series on Common Sense Knowledge, pp. 82–87. AAAI, Virginia (2010)
6. Cua, J., Ong, E., Manurung, R., Pease, A.: Representing Story Plans in SUMO. In: NAACL Human Language Technology 2010 Second Workshop on Computational Approaches to Linguistic Creativity, pp. 40–48. ACL, Stroudsburg (2010)
7. Liu, H., Singh, P.: Commonsense Reasoning in and over Natural Language. In: 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems. Springer, Berlin (2004)
8. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: FOIS, Ogunquit, Maine, October 17-19, pp. 2–9 (2001)
9. Lieberman, H., Liu, H., Singh, P., Barry, B.: Beating Common Sense into Interactive Applications. AI Magazine 25, 63–76 (2004)
10. Swartjes, I.: The Plot Thickens: Bringing Structure and Meaning into Automated Story Generation. Master's Thesis. University of Twente, The Netherlands (2006)
11. Ang, K., Yu, S., Ong, E.: Theme-Based Cause-Effect Planning for Multiple-Scene Story Generation. In: 2nd ICCC, Mexico City, Mexico, April 27-29, pp. 48–53 (2011)
12. Chen, J., Liu, J.: Disambiguating ConceptNet and WordNet for Word Sense Disambiguation. In: 5th IJCP 2011, Chiang Mai, Thailand, pp. 686–694 (2011)
13. Speer, R., Havasi, C., Lieberman, H.: AnalogySpace: Reducing the Dimensionality of Common Sense Knowledge. In: Cohn, A. (ed.) 23rd National Conference on Artificial Intelligence vol. 1, pp. 548–553. AAAI Press (2008)
14. Kipper, K.S.: VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. Ph.D. Dissertation. University of Pennsylvania, Philadelphia (2005)
15. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
16. Machado, J.: Storytelling. In: Early Childhood Experiences in Language Arts: Emerging Literacy. Thomson/Delmar Learning, New York (2003)
17. Ryan, M.: Cheap Plot Trick, Plot Holes, and Narrative Design. Narrative 17(1), 56–75 (2009)