

Using Synthetically Collected Scripts for Story Generation

Takashi Ogata

Iwate Prefectural University
/ 152-52, Sugo, Takizawashi,
Iwate, 020-0693 Japan
t-ogata@iwate-
pu.ac.jp

Tatsuya Arai

Iwate Prefectural University
/ 152-52, Sugo, Takizawashi,
Iwate, 020-0693 Japan
g0311011@s.iwate-
pu.ac.jp

Jumpei Ono

Iwate Prefectural University
/ 152-52, Sugo, Takizawashi,
Iwate, 020-0693 Japan
g236m001@s.iwate-
pu.ac.jp

Abstract

A script is a type of knowledge representation in artificial intelligence (AI). This paper presents two methods for synthetically using collected scripts for story generation. The first method recursively generates long sequences of events and the second creates script networks. Although related studies generally use one or more scripts for story generation, this research synthetically uses many scripts to flexibly generate a diverse narrative.

1 Introduction

A script, originally related to a type of schema in Gestalt psychology (Bartlett 1923), is a knowledge-representation method in cognitive science and artificial intelligence (AI) (Schank and Abelson 1977). The authors use a script as one of the methods or techniques to generate stories in an Integrated Narrative Generation System (INGS), an automated NGS architecture that is already operating through incremental development (Ogata 2016). Ogata, Arai, and Ono (2016) comprehensively introduced how to use a script in INGS. The script is organically positioned as one of the story-generation techniques, especially for detailed episodic sequences of events or a character's sequential actions.

This paper presents two methods for synthetically using collected scripts for story generation in INGS. The first method recursively generates long sequences of events and the second creates script networks. Related studies deal with one or more scripts for story generation. In contrast, this research synthetically uses many scripts to flexibly generate a diverse narrative. Although this paper does not discuss the semantic aspects of a script, we will add semantic mechanisms to the proposed formal methods in the future.

Kybartas and Bidarra (2016) classified 67 types of narrative-generation systems based on the degree of automatic generation in a story and components in the story. The degree is divided into five steps (the degree for story are “Manual”, “Structure”, “Template”, “Constrained”, “Automated”, and the degree for components are “Manual”, “Modification”, “Simulation”, “Constrained”, “Automated”). INGS is positioned as “constrained” (level 4 of the five steps) in both points of view. In the evaluation, a story and components are fully generated.

2 Scripts and INGS

Narratology (Prince 1982) divides a narrative into structural elements (story and discourse). A story refers to temporally ordered events. Though temporal order is a category of semantic techniques that organically combines events, the semantic mechanism includes other techniques types. A discourse means ordered events in which a story is narrated and also includes surface text; it is constructed using many techniques, e.g., causal relation and macro narrative structures. INGS is designed based on this idea (Figure 1). A narrative-generation process is conducted through the following mechanisms: story generation, discourse, and surface representation. INGS has knowledge mechanisms, including conceptual dictionaries (Ogata 2015) and language-notation dictionaries. Moreover, INGS has narrative-content knowledge bases to store basic fragmental, patternal, and structural knowledge. A detailed description of the current version can be found in (Ogata 2016).

This paper focuses on story generation. A story can be constructed at various levels of detail. “Taro eats sushi at a restaurant” can certainly be an event in a story. More detailed sequences of events for “eating at a restaurant” can also be elements in a story. Each of the scripts extends an event into a sequence of events to solidify or detail the process.

In related studies, a story grammar (Rumelhart 1975) hierarchically details a story from the macro structure to the micro one. A goal-plan (Newell and Simon 1972; Schank and Abelson 1977) details an event or a sequence of events using a planning action toward a character’s goal. Scripts, story grammars, and goal-plans are major structural techniques that solidify part of a story or the story itself. In INGS, the techniques for each generation and their order are not predetermined. Different techniques are flexibly, collectively, and co-operatively used. Additionally, though the *StoryNet* scripts by Singh, Barry, and Liu (2004) have a branch structure, INGS scripts have a simple pattern with no branches. Moreover, though Fujiki, Nanba, and Okumura (2002) acquire two terms per script, INGS has an unlimited number of terms in scripts.

A story in INGS is described using a hierarchical tree structure that uses relations to combine the sub-structures. At the lowest level, events and states are organized temporally. A script is also one of the relations. Techniques for story generation are called story techniques. They are basically defined by a story’s relations, and each relation uses the corresponding information in the story-content knowledge base. A script technique is also a story technique. Figure 2 shows the script technique mechanism. Each script is stored in the script-content knowledge base. Ogata, Arai, and Ono (2016) provide a detailed explanation.

INGS inputs a parameter that defines the story structure. All story techniques are selected based on this parameter. A story technique inputs a node from the story-tree structure and outputs a sub-tree that is structured by relations corresponding to the technique. In particular, a script technique outputs a sub-tree structure constructed of three or more events. A story tree is expanded by substituting the nodes in the sub-tree. In Figure 2, the “E2” node is substituted by the output sub-tree.

3 Collecting Scripts and their Synthetic Use in INGS

Various script-collection or acquisition methods are available. Manual acquisition directly writes scripts using rules and limitations, imagination and experience, or narrative analyses. Automatic acquisition has also been explored (Fujiki, Nanba, and Okumura 2002). Automatic organization in this paper means that new script knowledge is generated based on previously acquired scripts.

3.1 Collecting Scripts

Scripts were collected using the following process:

- 1) Two-hundred seventy-six university students freely wrote 873 natural language scripts based on simple examples, without special semantic constraints. We collected 860 scripts by checking each event’s flow.
- 2) We transformed the natural-language scripts into the corresponding INGS case structures using a semi-automatic script-description tool (Arai, Ono, and Ogata 2016). Specifically, a user decides the meaning of each verb concept in a script from candidates in the verb-conceptual dictionary. The tool checks the consistency using the case structures and stores the completed script in the script-content knowledge base.
- 3) We set each script’s name to correspond to a verb-concept name included in the verb-conceptual dictionary.

The original 860 script names overlapped extensively. Only 332 script names remained when the overlapping ones were removed. The verb-conceptual dictionary includes 11,951 verb concepts. About 2.78% of the verb concepts have been scripted. In the future, we intend to convert all the verb concepts to scripts.

3.2 Recursively Combining the Collected Scripts

A script is expanded by a recursive combination process from the first element (Figure 3). If the verb concept in a script event equals the name of another script, the original script is expanded by the latter script. We cannot repeat a previously used script. Table 1 shows the result of an experiment.

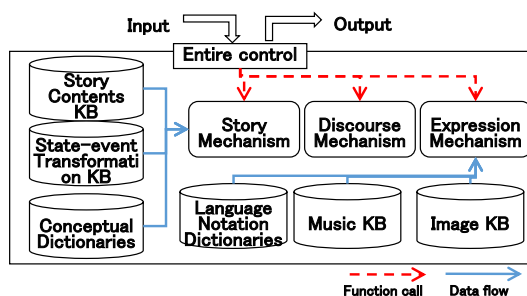


Figure 1. Overview of INGS Architecture

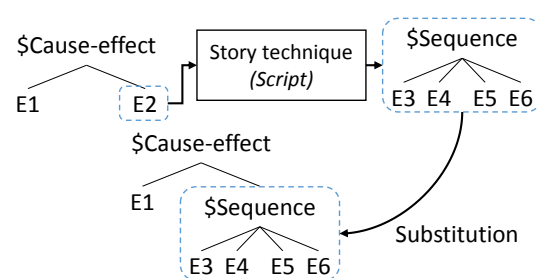


Figure 2. IO Structure of Script Technique

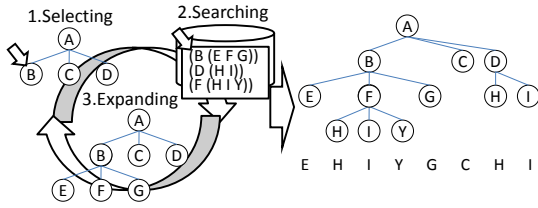


Figure 3. Recursive Script Combination

Table 1. Experimental Results of Script Combination

Timing	Script length		
	Min	Max	Average
Before	2	15	6.85
After	2	190	43.96

3.3 Generating Script Networks using the Collected Scripts

We generated script networks using the collected scripts. Basically, the next script for each script is based on a verb concept. In particular, this method pairs all scripts in a temporal order. If overlapping pairs for a script overlap, only one pair is used. Figure 4 shows the four-step process with a concrete example. *Cytoscape* is an open-source software product for visualizing network graphs by the U.S. National Institute of General Medical Sciences.

Figure 5 (left) shows the entire generated script networks that including five networks. Figure 5 (right) zooms in on a part of the main network that includes 1127 verb concepts. The “average path length” in Table 2 means the average distance between any two nodes. The dispersion in the values’ variance is relatively small. Table 3 shows the characteristics of elements in the networks. The “starting node” and “terminal node” respectively mean the arrows from a node and to a node. If the “harmonic mean” value is higher, the verb concept appears more frequently in the starting point, middle points, and terminal point in a script.

3.4 Using Generated Scripts for Story Generation in INGS

We present an overview of using synthesized scripts for story generation. There are two methods—1) recursive generation and 2) script networks—and two techniques—a) detailing and b) inclusion—for a total of four script techniques (1-a, 1-b, 2-a, and 2-b). The former technique (a) expands or substitutes an event in a story by a sequence of detailed events, and the latter (b) expands or substitutes an event in a story by a sequence in which the event is included (Figure 6).

In addition, we insert adequate values into each event case in all scripts using a semi-automatic script-description tool and give a name to each script. The case values are associated with the conceptual dictionaries in INGS. As many scripts are very long, part of a script can be cut to make it shorter. For example, the average number of events in 50 “have-a-meal1” scripts is 32.

1-a: A target event in a story is expanded by detailing it with a script sharing the name of the verb concept in the event.

1-b: A target event in a story is expanded using a script in which the verb concept of the target event is included. Figure 7 shows an example of a script structure generated by this method.

2-a: A target event in a story is expanded by substituting part of it with a script in the network that

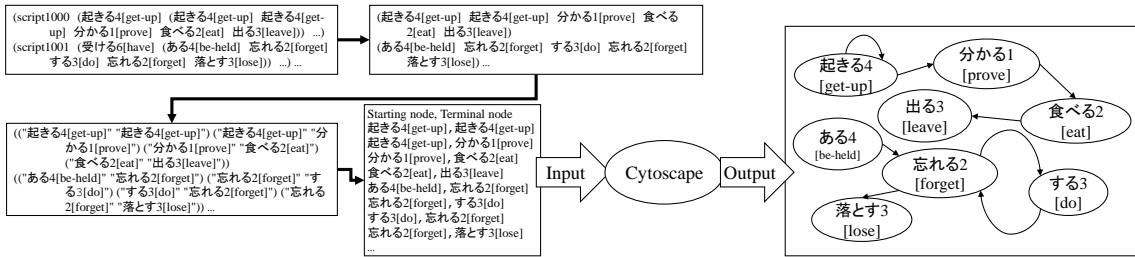


Figure 4. Script-Network Generation Process

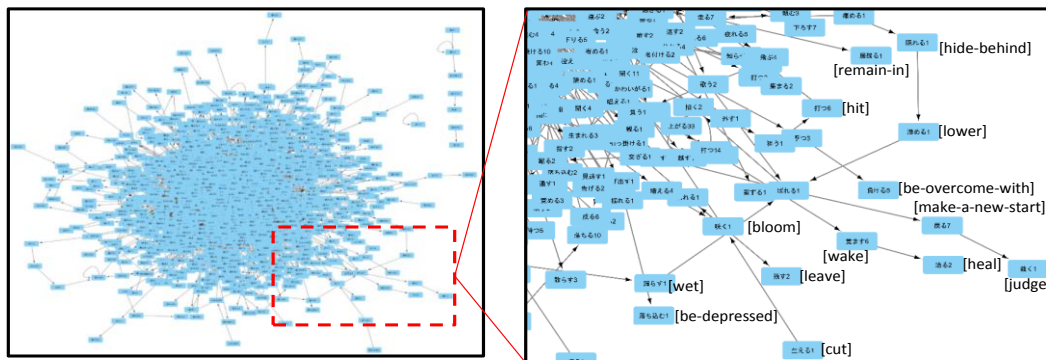


Figure 5. Five Script Networks and the Focused Part

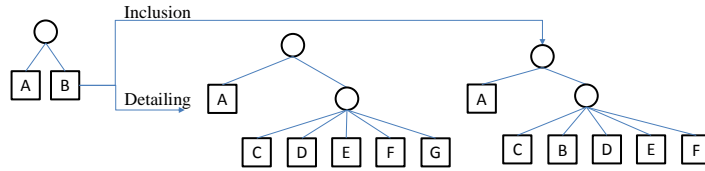


Figure 6. Using the Combined Script

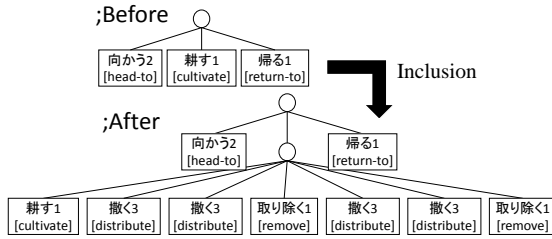


Figure 7. Generated Structure (1-b)

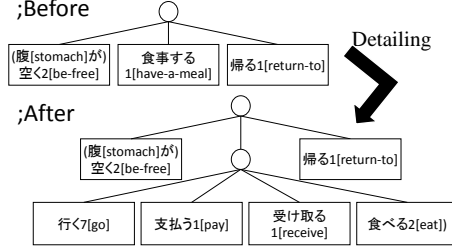


Figure 8. Generated Structure (2-a)

shares the name of the verb concept in the event. Figure 8 shows an example of the structure generated using this method and Figure 9 shows the script.

2-b: A target event in a story is expanded by using a script in which the verb concept in the target event is included.

Before	(\$継起[sequence]) (event 空く 2[be-free] (type action) (ID 1) (time (time1 time2)) (agent age%青年 [young-man]#1) (location loc%家 [house]#1) (object obj%腹 [stomach]#1)) (event 食事する 1[have-a-meal] (type action) (ID 2) (time (time2 time3)) (agent age%青年 [young-man]#1) (location loc%家 [house]#1)) (event 帰る 1[return-to] (type action) (ID 6) (time (time6time7)) (agent age%青年 [young-man]#1) (from loc%料理屋 [restaurant]#1) (to loc%家 [house]#1)))
	(\$継起[sequence]) (event 空く 2[be-free] (type action) (ID 1) (time (time1 time2)) (agent age%青年 [young-man]#1) (location loc%家 [house]#1) (object obj%腹 [stomach]#1)) (\$継起[sequence]) (event 行く 7[go] (type action) (ID 2) (time (time2 time3)) (agent age%青年 [young-man]#1) (location loc%家 [house]#1) (object obj 買い物 [shopping]#1)) (event 支払う 1[pay] (type action) (ID 3) (time (time3 time4)) (agent age%青年 [young-man]#1) (counter-agent age%店員 [clerk]#1) (location loc%料理屋 [restaurant]#1)) (event 受け取る 1[receive] (type action) (ID 4) (time (time4 time5)) (agent age%青年 [young-man]#1) (location loc%料理屋 [restaurant]#1) (object obj%ハンバーグ [hamburger]#1) (from age%店員 [clerk]#1)) (event 食べる 2[eat] (type action) (ID 5) (time (time5 time6)) (agent age%青年 [young-man]#1) (location loc%料理屋 [restaurant]#1) (object obj%ハンバーグ [hamburger]#1)) (event 帰る 1[return-to] (type action) (ID 6) (time (time6time7)) (agent age%青年 [young-man]#1) (from loc%料理屋 [restaurant]#1) (to loc%家 [house]#1)))
After	

Figure 9. Generation Example (2-a)

Table 2. Script Network Characteristics (1)

	Number of verb concepts	Average path length	Variance		Number of verb concepts	Average path length	Variance
Network1	112/	4.30	0.33	Network4	2	1.00	0.00
Network2	3	1.33	0.02	Network5	1	0.00	0.00
Network3	2	1.00	0.00				

Table 3. Script Networks Characteristics (2)

Verb concept	Starting node	Terminal node	Harmonic mean	Verb concept	Starting node	Terminal node	Harmonic mean
食べる 2[eat]	148	182	163.25	死ぬ 1[die]	0	5	0.00
出る 3[leave]	116	152	131.58	寛ぐ 1[relax]	0	3	0.00
行く 7[go]	144	91	111.52	盛る 3[incorporate]	0	3	0.00
乗る 1[ride]	112	104	107.85	悲しむ 1[feel-sad]	0	2	0.00
洗う 2[wash]	79	77	77.99	出す 12[give]	0	2	0.00
見る 2[visit]	64	67	65.47	起きる 2[break-out]	4	0	0.00
買う 2[incur]	70	57	62.83	減る 2[run-short]	3	0	0.00
帰る 2[come-home-from]	44	97	60.54	思う 5[think]	3	0	0.00
行く 3[go-from]	70	44	54.04	張る 3[fill]	2	0	0.00
選ぶ 1[choose]	50	47	48.45	惚れる 1[fall-in-love]	2	0	0.00

4 Conclusions

INGS included two methods for using synthetically collected scripts for story generation. The first recursively generated a long sequence of events and the second created script networks. This paper implemented both methods and showed their effectiveness in the INGS architecture through actual generated examples. Future issues include semantic consideration, automated script acquisition, etc.

Reference

- Arai T., Ono J. and Ogata T. 2016. Semi-automatic generation of events sequence knowledge for narrative generation: The use in an integrated narrative generation system, *Proc. 30th Annual Conf. of the Japanese Society for Artificial Intelligence*, 3P1-1in2.
- Bartlett F. C. 1923. *Psychology and Primitive Culture*, Cambridge University Press, England.
- Fujiki T., Nanba H. and Okumura M. 2002. Automatic acquisition of a script knowledge from a text collection, *Proc. of the Forum on Information Technology*, 2002 (2): 123–124.
- Kybartas B. and Bidarra R. 2016. A survey on story generation techniques for authoring computational narratives, *IEEE Transactions on Computational Intelligence and AI in Games*, 99.
- Newell A. and Simon H. A. 1972. *Human Problem Solving*. Prentice Hall, UK.
- Ogata T. 2015. Building Conceptual Dictionaries for an Integrated Narrative Generation System, *Journal of Robotics, Networking and Artificial Life*, 1 (4): 270–284.
- Ogata T. 2016. Computational and cognitive approaches to narratology from the viewpoint of narrative generation, Ogata T. and Akimoto T. (Eds.). *Computational and Cognitive Approaches to Narratology*, USA: IGI Global Publishing, 1–73.
- Prince G. 1982. *Narratology*, Walter de Gruyter & Co., Berlin.
- Rumelhart D. E. 1975. Notes on a schema for stories, Bobrow D. G. and Collins A. M. (Eds.), *Representation and Understanding: Studies in Cognitive Science*, Academic Press, The Netherlands.
- Schank R. C. and Abelson R. P. 1977. *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum, NJ.
- Singh P., Barry B. and Liu H. 2004. Teaching machines about everyday life, *BT Technology Journal*, 22 (4): 227–240.