Learning by Aligning Stories

Zhutian Yang,[1] Patrick Henry Winston[2]
[1]Nanyang Technological University
[2]Computer Science and Artificial Intelligence Laboratory, MIT
and Center for Brains, Minds, and Machines, MIT

Abstract

We describe how the Genesis Story-Understanding System acquires knowledge by aligning successful problem-solving stories. Two stories told about repair each may involve a great deal of irrelevant detritus. To extract a common repair recipe, Genesis must determine which story elements are fundamental to solving the problem and which are incidental. Genesis's overall strategy is to identify the sub-goals in each story by abduction from specified actions and then align the sub-goals to construct recipes.

"History repeats itself. "

"He made the same mistake twice. "

"The movie is a modern retelling of Romeo and Juliet."

Imagine that you heard these sentences not from a human, but from a computer program. You would surely ask yourself how a program could read two different-looking stories and then discover common elements in them. We think answering that question is important because we believe the ability to discover common elements enables humans to learn from experience, to embrace what is relevant in solving problems, and to avoid what gets in the way.

In this paper, we describe how the Genesis Story-Understanding System learns common problem-solving steps by aligning successful problem-solving stories. We demonstrate how Genesis learns to replace a phone battery from two 80-word stories that have much irrelevant detail and nothing expressed in exactly the same way.

The key idea is to align different-looking steps in two stories by noting the goals they achieve.

First, Genesis identifies sub-goals in each story by abduction from specified actions. For example, Genesis aligns *Alice removed the cover of the phone* with *Bob took the cover off the phone* because both steps achieve the goal of *exposing the phone's parts* according to common sense abduction rules. Given these two steps align, we can assume the goal they achieve to be important. In contrast, *Bob poured himself a cup of coffee* is considered incidental if the goal it achieves appears only in Bob's story.

Then, Genesis constructs recipes based on those sub-goals that align. One is an abstract recipe listing the sub-goals; two others are specific recipes, based on using common sub-goals to identify the relevant parts of the two contributing stories.

The rest of the paper is organized as follows: We first explain what we mean by learning, stories, and problem solving, and we introduces how they are modeled in the Genesis Story Understanding system. Next, we demonstrate a simple *Replace phone battery* example to illustrate the learning mechanism. Lastly, we conclude with our contributions.

Story-Enabled Learning

Alan Turing, in *Computing Machinery and Intelligence*, wrote about what he called the "education" and "other experience" of a child machine (1950). Marvin Minsky took the idea to another level in *The Emotion Machine* when he wrote about how children analyze and compare stories they tell themselves (2006).

Our work is in that tradition. Because much of education and experience is about stories, we believe a model of human intelligence should include an explanation of how we learn by analyzing stories, both as children and as adults.

We define stories as sequences of complex, nested descriptions of entities, relations, and events. Stories include accounts of what we observe, what happened, what we believe, and what to do. Because most of our everyday problem-solving processes involve sequences of actions, this work follows the hypothesis that problem-solving is a special case of following recipes that tell us what to do and recipe following is a special case of story understanding (Winston 2018).

The Genesis System models how humans understand and construct stories (Winston and Holmes 2018). In the work reported here, we extend the Genesis System beyond, for example, summarizing, comparing, inferring from, and answering questions about stories, to discovering what-to-do knowledge from stories as in the answer to "How do you replace the battery of a phone?" The learning problem is: how can a program generate recipes, which are sequences of what-to-do actions aimed at solving a problem, by reading pairs of stories expressed in English.

Learning by Aligning Stories

Consider the first story about replacing a phone battery:

> Start story titled "Alice's story." Alice put her pants in the washing machine. Alice forgot to take her phone from her pants. The phone became wet. It does not work. Alice quickly removed the cover from the phone and she collected the old battery from the phone. Alice purchased the replacement battery online. The replacement battery arrived and she inserted it into the phone. She put the cover on the phone. She recharged her phone for four hours. Bravo! The phone worked again. The end.

What can be learned from the story about replacing a phone battery? Not that you have to put your pants in a washing machine. Not even that the phone became wet. Irrelevant details swamp the key elements of phone battery replacement. How can a program decide which details to include in a phone battery replacement recipe?

The problem should be easier if there is a second story, such as this one:

> Start story titled "Bob's story." A client's phone did not work. The client asked Bob to change the battery of the client's phone. Bob made coffee and drank it. Bob got a replacement battery from the stockroom. Bob slid down the cover from the client's phone. Bob takes out the old battery from the client's phone and put the replacement battery in the phone. Lastly, he placed the cover on the client's phone. The client's phone started working again. The client's problem is solved. The client was happy. The end.

One idea would be to keep only those parts that correspond, but alas, no parts correspond exactly. You could argue that there are paraphrases and paraphrases indicate correspondence, but how can you determine which elements are paraphrases?

Our approach deploys the full power of our Genesis story understanding system. Details and references to related work are reported in detail elsewhere

- Genesis analyzes the stories producing elaboration graphs using rules and concept patterns.
- Genesis notes where various actions imply common sub-goals by abduction.
- Genesis forms a general recipe from the common sub-goals in the two stories.
- Genesis forms two phone-specific recipes by listing the actions that produced the common sub- goals.

Genesis identifies common sub-goals with abduction rules

Here are the abduction rules to determine the sub-goals indicated by various actions:

```
xx and yy are persons.
If xx drinks coffee, then xx must want to
 ↪  become alert.
If xx puts xx's yy in a washing machine,
 ↪  then xx must want to clean yy.
If xx removes the cover from yy, then xx
 ↪  must want to expose yy's parts.
If xx slides down the cover from yy, then xx
 ↪  must want to expose yy's parts.
If xx collects yy from zz, then xx must want
 ↪  to separate yy from zz.
If xx takes out yy from zz, then xx must
 ↪  want to separate yy from zz.
If xx inserts yy into zz, then xx must want
 ↪  to incorporate yy into zz.
If xx put yy in zz, then xx must want to
 ↪  incorporate yy into zz.
If xx puts the cover on yy, then xx must
 ↪  want to attach the cover to yy.
If xx places the cover on yy, then xx must
 ↪  want to attach the cover to yy.
```

We note that many of the words used in the wanting specifications evoke the container metaphor of Lakoff and Johnson (1980). This is perhaps not surprising in part-replacement scenarios.

In its analysis, Genesis also uses various common sense deduction and explanation rules (marked with may) that enrich understanding and generate elements that somehow must be ignored:

```
If xx forgot to take yy from zz, then zz
 ↪  contains yy.
If a phone becomes wet, then the phone may
 ↪  not work.
If xx's problem evidently is solved, then xx
 ↪  may be happy.
```

Now, when Genesis reads Alice's story, Genesis produces the elaboration graph shown in Figure 1.

When Genesis reads Bob's story, Genesis produces the elaboration graph shown in Figure 2. The abductions enable Genesis to discover instances of the Action-evoking goal concept pattern:

```
Start description of "Action-evoking goal".
xx's wanting vv leads to aa.
The end.
```

At this point, the abducted sub-goals in two Action-evoking goal patterns do not match:

```
Alice wants to clean her pants.
Bob wants to become alert.
```

Four do match:

```
Alice wants to expose her phone's parts.
Bob wants to expose the client's phone's
 ↪  parts.

Alice wants to separate the old battery from
 ↪  her phone.
Bob wants to separate the client's phone's
 ↪  battery from the client's phone.
```

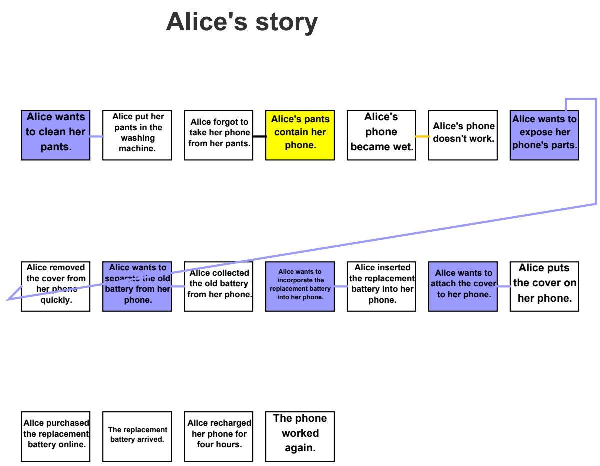

Figure 1: Alice's story. Abducted sub-goals appear in purple. A deduction appears in yellow. An explanation rule, signaled by an orange line, connects getting wet with not working.

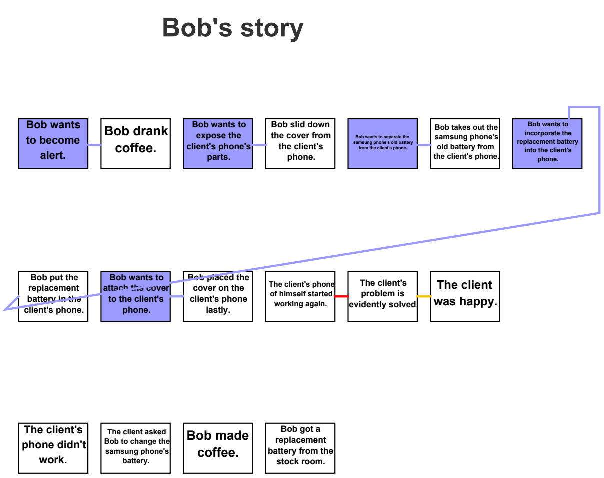

Figure 2: Bob's story. Abducted desires appear in purple. An explanation rule, signaled by an orange line, connects a solved problem with a client's happiness. A post-hoc-ergo-propter-hoc clausal connection is inferred from the juxtaposition of the final sentences in the story.

```
Alice wants to incorporate the replacement
 ↪ battery into her phone.
Bob wants to incorporate the replacement
 ↪ battery into the client's phone.

Alice wants to attach the cover to her
 ↪ phone.
Bob wants to attach the cover to the
 ↪ client's phone.
```

Genesis creates recipes from common sub-goals

From the matches, Genesis creates a recipe consisting of four goals from the objects of the wanting expressions. The format conforms to the way recipes are rendered for Genesis's story-grounded problem solver (Winston 2018).

Abstract battery replacement recipe:

```
Step: Expose the phone's parts.
Step: Separate the old battery from the
 ↪ phone.
Step: Incorporate the replacement battery
 ↪ into the phone.
Step: Attach the cover to the phone.
The end.
```

Looking at the other end of the matching abductions, the consequences, Genesis also creates two recipes, each of which works for a particular phone. Note that Genesis removes entity owners from the inner-language expressions before translating them to English:

Specific battery replacement recipe learned from Alice's story:

```
Step: Remove the cover from the phone.
Step: Collect the old battery from the
 ↪ phone.
Step: Insert the replacement battery into
 ↪ the phone.
Step: Put the cover on the phone.
The end.
```

Specific battery replacement recipe learned from Bob;s story:

```
Step: Slide the cover from the phone.
Step: Take out the old battery from the
 ↪ phone.
Step: Put the replacement battery in the
 ↪ phone.
Step: Place the cover on the phone.
The end.
```

Contributions

- We explained how the Genesis System can discover problem-solving recipes by aligning problem-solving stories, finding implied sub-goals, and then constructing steps from common sub-goals.
- We demonstrated learning by aligning stories by having the Genesis System create abstract and specific recipes for replacing a phone battery from two different-looking stories told by humans.

Acknowledgments

The work described in the paper was supported, in part, by the Air Force Office of Scientific Research, contract number FA9550-17-1-0081. The battery replacement task was suggested by Dynamic Object Language Labs, Inc.

References

Lakoff, G., and Johnson, M. 1980. *Metaphors we live by*. University of Chicago Press.

Minsky, M. 2006. *The Emotion Machine*. New York, NY: Simon and Schuster.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59(236):433.

Winston, P. H., and Holmes, D. 2018. The Genesis Enterprise: Taking artificial intelligence to another level via a computational account of human story understanding. MIT DSpace, Computational Models of Human Intelligence Community, CMHI Report Number 1. URL `http://hdl.handle.net/1721.1/119651`.

Winston, P. H. 2018. Self-aware problem solving. MIT DSpace, Computational Models of Human Intelligence Community, CMHI Report Number 2. URL `http://hdl.handle.net/1721.1/119652`.