Plotter: Operationalizing the Master Book of All Plots

Markus Eger, Colin M. Potts, Camille Barot, R. Michael Young
Liquid Narrative Group, Department of Computer Science
North Carolina State University, Raleigh, NC 27695
{meger, cmpotts, cbarot}@ncsu.edu, young@csc.ncsu.edu

Abstract

Pulp fiction author William Wallace Cook published *Plotto: The Master Book of All Plots* in 1928, which contains almost 2000 plot fragments and relatively formal instructions on how human authors could combine them to produce plots behind novels. In this paper we show one way that the methods in this book can be used to computationally generate plots from the fragments. We also show sample plots generated by our system called *Plotter* that uses this method. Finally we use them to discuss idiosyncrasies and limitations of the book.

Introduction

The generation of stories by a computer has been the subject of research for many years starting with TaleSpin (**?**) and still ongoing in efforts by a number of researchers around the world. Current approaches either work with a knowledge-base that was acquired from existing artifacts, or use domains created specifically for the purpose of story generation. The former approach requires access to a corpus in machine-readable and -understandable format, while designing domains for the latter approach is non-trivial. However, some human authors of stories have taken the time to write down their approach to plot-generation in the form of books. One of these books, *Plotto: The Master Book of All Plots* was published in 1928 by prolific pulp fiction author William Wallace Cook (**?**). While the title may seem pompous, the book does cover a wide variety of different plot elements. What makes it interesting for computational purposes is that it is written in a very structured way: It contains 1852 abstract plot fragments, each of which is identified by a number and refers to several potential predecessors and successors. For example, plot fragment 1068 reads as follows:

```
(384) (908) B comes to understand the
evil of her selfish outlook upon life
when one of her children, CH, dies
(581) (1069 ch D-A to CH) (1240)
```

In Cook's notation, this indicates that the fragments 384 and 908 are potential predecessors, and 581, 1069 and 1240 are potential successors to this fragment. As can be seen, there may be modifiers applied to the following segments, as in `ch D-A to CH`, which means "change all occurrences of D-A (daughter of A) to CH (CH-B, child of B) in the following segments." We discuss these modifiers and other special cases in more detail in the following sections.

The goal of this paper is two-fold: First, it introduces the book *Plotto: The Master Book of All Plots* to parts of the story generation community, describes how it is intended to work for human authors and details how we operationalized this approach. Second, it also describes the general approach of story generation as a random walk on a graph, for which *Plotto* is just one possible data source.

Related work

The idea to use existing work on the structure of stories to generate them is not new. One of the most popular narratology work is that of Vladimir Propp, which analyzes the structure of Russian folk tales (**?**). Propp describes the structure of folk tales to consist of 31 "functions" whose appearance and order in a given story are constrained by a set of rules. Several authors have turned Propp's analytical work into generative systems (e.g., (**?**), (**?**)).

Another seminal work is Georges Polti's "The Thirty-Six Dramatic Situations" (**?**), in which he describes the eponymous 36 dramatic situations that, he claims, all plots consist of. As opposed to Propp, Polti does not propose a theory to combine the different situations into a global structure. Still, this work can be used in the generation of narratives, as demonstrated by Jhala and Young's Darshak system (**?**), in which the dramatic situations are used as patterns for creating a cinematic visual discourse of an existing plot.

Other systems are based on plot grammars or plot graphs directly extracted from story corpora, e.g., old French epics (**?**), or crowdsourced stories (**?**).

Plot generation with *Plotto*

Our approach to plot generation is based on "Plotto: The Master Book of Plots" (**?**). It differs significantly from Propp's and Polti's work in granularity. Whereas Propp and Polti described very few high-level concepts, *Plotto* contains 1852 different plot fragments that are relatively specific. For example, plot fragment 1457 in *Plotto* reads

```
A, accused by A-9 of having stolen
a valuable document, X, has his
```

```
innocence proved when X is found
between the leaves of a book which
A-9 had been reading
```

which is just one possible way Propp's "Exposure" function could be implemented in a story.

Plotto structure

As already mentioned, *Plotto* contains 1852 numbered story fragments. Each of these contains possible predecessor and successor fragments, and may make reference to any of the 54 different character symbols, including the male protagonist `A`, the female protagonist `B`, a male employer `A-9`, a mysterious object `X`, and so on. Successor fragments can be associated with additional instructions on how they have to be changed to fit with the current fragment, e.g., by exchanging one character for another. For example, one of (previously mentioned) fragment 1457's successors is `1337 ch A-8 to A-9`, i.e. fragment 1337 with the character `A-8` replaced by the character `A-9`. Fragment 1337 reads:

```
A stumbles over the body of a
murdered man, A-8
```

By replacing `A-8` (male utility symbol) by `A-9` this fragment changes from being completely unrelated to the story so far to being more dramatic, because `A` finds his accuser `A-9` to be dead.

Plotto defines several basic modifiers:

- `ch` ϕ `to` ψ: Change all occurrences of character ϕ to character ψ.
- `tr` ϕ `&` ψ: Exchange the characters ϕ and ψ.
- `-*`, where the successor fragment contains an asterisk: Only include text up until the asterisk. Variants may include multiple asterisks, e.g. `-**`, and a "starting asterisk", e.g. `*-***`, which means "only include text from the first to the third asterisk".
- A fragment can also have multiple successors, separated with semicolons, which means that they should all follow in order before allowing further choices by the author.

Additionally, there are a few "one off" modifiers like `without the clause in parenthesis` that have to be dealt with on a case-by-case basis.

Each fragment is categorized in a typology of 61 fragment types, referred to as B clauses. B clauses correspond to high-level descriptions of conflict situations, such as:

```
(40) EMBARKING UPON AN ENTERPRISE IN
     WHICH ONE OBLIGATION IS OPPOSED
     BY ANOTHER OBLIGATION
```

Alongside these 61 B clauses, *Plotto* also specifies a set of A clauses — character traits introducing the protagonist (e.g., `An Erring Person`) — and C clauses — high-level descriptions of situations that conclude the plot, for example `Undertakes a role that leads straight to catastrophe`. *Plotto* defines a "masterplot" as a combination of A clauses, a combination of B clauses, and a resolving set of C clauses. Given a masterplot, a set of fragments can be chosen to produce a plot. The masterplot sets the theme and restricts the set of available fragments. There are no rules about combining A, B, and C clauses, but the book includes many suggested combinations.

The intention behind the book was to provide human authors with inspiration for plots, not fully-fleshed out stories. Therefore the plot fragments are very high-level descriptions of what is supposed to happen in that part of the story.

Generation process

The generation process we define only generates a combination of B clauses, as these are the only elements that contribute to the plot itself. While *Plotto* contains the suggestion that an author may guide their plot generation process by choosing a subset of B clauses and only picking fragments corresponding to those clauses, our system always starts with all B clauses and therefore all fragments, because we want to explore the full range of plots that can be generated by *Plotto*. For the actual generation process, we parsed the contents of the book into a graph. For each B clause fragment in the book, we created a node in our graph labeled with the fragment's number. For every possible successor fragment, we created a directed edge from the source fragment to the possible successor. Each edge is labeled with the modifiers corresponding to that successor relation. Figure 1 shows an induced subgraph on the nodes shown.

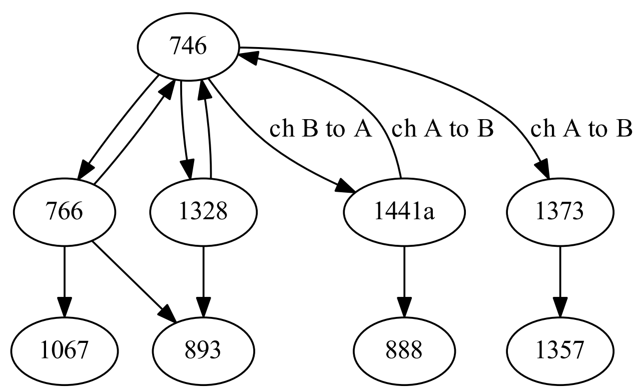

Figure 1: A small part of the graph constructed from *Plotto* plot fragments

Our plot generation process works by picking a starting node (based on the user's choice of category, of a specific node, or on a random node selection) and then following random edges, i.e. performing a random walk on the graph. If a traversed edge has a non-empty label, the modifier corresponding to that label is added to a modifier list and applied to all subsequent plot fragments, if applicable. For example, when starting at node 746 in Figure 1, the algorithm can pick node $1441a$ as the next node, but it then would have to change all occurrences of character `A` to character `B`. If it then proceeds to node 888, it would have to replace character `A` with character `B` in that plot fragment as well.

This generation process continues until a stopping criterion is reached. Because Cook didn't provide a specific stopping criterion, we implemented three different ones:

- stop upon reaching a node that has no possible successors,
- stop upon visiting a user-defined maximum number of plot fragments,
- stop by random chance according to a user-defined probability.

The generated plot then consists of all visited plot fragments in order. Figure 2 shows the generation algorithm. The functions `fragment` and `modifiers` return a fragment's text or modifiers for a particular fragment transition (edge), respectively, while the function `apply` applies a modifier to a fragment's text. The symbol $\mathbb{1}$ is used for the closed real interval from 0 to 1, and the function `random` chooses a random element of a set. Note that this generation process is not tied to *Plotto*'s content itself, and it is possible to author other graphs that can be used by it.

Require: Story Fragment Graph (V, E)
Require: Functions fragment : $V \mapsto \textit{string}$, modifiers : $E \mapsto \textit{Modifiers}$
Require: Termination probability $\textit{termprob}$ and maximum plot length $\textit{sizelimit}$
  $\textit{plot} \leftarrow$ ""
  $\textit{state} \leftarrow \text{random}(V)$
  $\textit{mods} \leftarrow []$
  loop
    $\textit{text} \leftarrow \text{fragment}(\textit{state})$
    for all $\textit{mod} \in \textit{mods}$ **do**
      $\textit{text} \leftarrow \text{apply}(\textit{text}, \textit{mod})$
    end for
    $\textit{plot} \leftarrow \textit{plot} + \textit{text}$
    $\textit{successors} \leftarrow \{(a, b) \in E \mid a = \textit{state}\}$
    if $\textit{successors} = \varnothing$ **or** $|\textit{plot}| = \textit{sizelimit}$ **or** $\text{random}(\mathbb{1}) \leq \textit{termprob}$ **then**
      break
    end if
    $(\textit{current}, \textit{state}) \leftarrow \text{random}(\textit{successors})$
    $\textit{mods} \leftarrow \textit{mods} + \text{modifiers}((\textit{current}, \textit{state}))$
  end loop
  return $\textit{plot}$

Figure 2: The plot generation algorithm

Discussion

Sample plots

To illustrate the kind of plots that can be expected from our generation process, Figures 3 and 4 show plots generated by Plotter. In both plots, the parenthesized numbers at the start of each paragraph refers to the plot fragment ID from *Plotto*, and are just included for reference. As can be seen in Figure 3, many plot fragments contained in *Plotto* are very vague, as they are only meant to give inspiration to an author, like the unspecified "misfortunes." Some, however, are very specific, like fragment $1438c$ in Figure 4, with the line "One Bricktop for all, all Bricktops for one." Generally, though, each plot fragment is not to be seen as actual text for inclusion in the story, but rather as a high-level overview of what should be conveyed in this part of the story.

(900) A's imagination leads his mind astray, and in seeking emancipation from fancied misfortunes he is plunged into real misfortunes. A, given all he thought necessary for his happiness, finds there is still something lacking - something in himself.
(1348a) A's mortal pride would transgress the Divine Power; so fate, seeking to discipline A, materializes a spirit, AX, in A's image. A's property and high place in the world are appropriated by AX, and when A would claim his earthly possessions, he is treated as an imposter.
(715) A, wealthy and influential, loses his clothes, personal belongings and all other means of identification.
(658) A, a man of high standing in his community, fears that through unusual conditions his character will be discredited. A seeks to safeguard his reputation, which is threatened by unusual conditions.
(652b) A, the idol of his people, is about to lose a wrestling match to A-3, his rival. A, wrestling in the open with A-3, a rival wrestler, prevents A-3 from winning a victory by falling from a cliff, apparently by accident, and losing his life.

Figure 3: A sample plot generated by Plotter

Limitations

Limitations of this story generation process are twofold: Some are inherent to its general approach to story generation where each step only depends on the current state of the story and not past states, but others are specific to *Plotto*.

The main limitation of the approach is the generator's lack of memory. For example, in Figure 4 above, character `A` thinks that he has killed his friend `A-2`, in fragment 1344, but then is surprised to learn that the friend is still alive in fragment 794. In the next fragment, however, `A` still thinks that `A-2` is dead, and has in fact been murdered by `A-5`. Without any additional explanation, it is hard to justify why `A` would change from thinking that he murdered `A-2` himself to believing that `A-5` was the murderer, especially after he has seen that `A-2` is actually still alive. It should also be noted that `A` is introduced to be in a secret society, and later on is a detective, while the whole incident that kicked off the story, `B`'s suicide, is never referred to again, nor is the stranger `A-4` that denies her help.

One way around this problem would be to have a tree-like structure for the graph, where each path from the root to a branch is consistent. The drawback of that approach is that it greatly reduces re-use of nodes in different story lines. The graph that is described by *Plotto*, on the other hand, contains many overlapping story lines and even loops. For example, the part of the graph shown in Figure 1 has multiple loops, all of which contain node 746. That node reads:

```
B, who was thought by the people of her
community to have supernatural powers,
is discovered to have been insane - a
condition caused by a great sorrow
```

Node 766, which is bidirectionally connected to 746 reads:

(1025) B is carrying out a secret enterprise and falls into danger. B, carrying out a secret enterprise and falling into danger, appeals for aid to a stranger, A-4; but A-4 is wary and refuses assistance.
(1417) B disappears, apparently a suicide when, in desperate danger, she implores aid and it is denied.
(802b) A discovers that B, supposed to have committed suicide, was really murdered.
(1154a) B, mistaken for another woman who is under ban of death by a strange secret society.
(1438c) A is a member of a strange secret society in which all the members are bound by oath to avenge the wrongs, real or fancied, of each individual member. A is a member of a strange secret society in which all members have red hair. The watchword of the society: "One Bricktop for all, all Bricktops for one".
(1319a) A, in order to be revenged upon his enemy, A-3, manufactures an infernal machine, X. A sends an infernal machine, X, to his enemy, A-3, and it falls into the hands of A's friend, A-2.
(1344) A has a delusion that he is haunted by A-2. A believes that the blood of A-2 is on his hands.
(794) A supposes his friend, A-2, is dead. A is astounded when his friend, A-2, whom he supposed to be dead, suddenly appears before him.
(1371) A is positive, in his own mind, that his friend, A-2, has been murdered by A-5. A suspects A-5, seemingly an honest man, of a crime, and tries to prove him guilty.
(1413b) A, a detective, unmasking A-5 as the leader of a criminal gang, finds that he cannot secure A-5's arrest as the police authorities refuse to act. A discovers that A-5 is a government secret service man, merely posing as a criminal in order that he may secure an advantage in prosecuting his work for law and order.

Figure 4: Another sample plot generated by Plotter

```
B, a plain woman, believes herself
surpassingly beautiful. B has a
character weakness which proves a
bar to many of her enterprises.
```

Both of these nodes provide a description of `B`'s background. The meaning of the loop is therefore not that a story should go back and forth between them, but rather that they can be used in either order, and the story continues from whichever one was visited last. Plotter supports this intention by preventing any node from being visited multiple times.

Another limitation of our work is that the generated artifact is just a plot, and not a finished text representing the story. The descriptions provided by *Plotto* are really meant as a guideline for authors, and would need to be fleshed out to be considered an actual story.

Conclusion and Future work

We have discussed *Plotto*, a book designed by its author to be used by human readers in the process of plot generation. The book contains 1852 numbered plot fragments, each with possible successor fragments. We have parsed this content into a graph data structure with each plot fragment being a node with directed edges to each of its successors. We then described our plot generation system *Plotter* that generates plots from this graph data structure. The plots generated by *Plotter* have some inherent limitations, since the possible choices available at each node depend only on that node, and not on anything that happened before in the story. We showed two sample plots generated by our system, and discussed how the limitations of the approach manifest themselves in those plots.

The consistency of generated plots could be facilitated by an AI planning operator representation of each fragment. This would require determining the necessary preconditions of each fragment along with its (possibly conditional) effects. This would allow a planning or logical system to validate the consistency of walks through the graph. Additionally, having a formal representation of the nodes would simplify the generation of an actual story text, which is currently non-trivial, and enable the system to determine what would be a good ending point for the story, as opposed to requiring the user to specify arbitrary thresholds.

Acknowledgements

This material is based upon work supported in whole or in part with funding from the Laboratory for Analytic Sciences (LAS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the LAS and/or any agency or entity of the United States Government.