Introduction

AGAST is an attempt to produce a program that can write intelligent stories. With an eclectic combination of ideas from the work of both computer scientists and writers, we have produced the flexible core of what could be a very intelligent story teller.

Work being done in cognitive science, natural language processing, and other areas that are closely based on human actions fits neatly into the project. Story-telling is not one isolated behavior, but a combination of many; work from text generation, decision-making, story-planning, character development and other areas is needed.

AGAST uses a formula similar to those developed by various professional writers to teach beginning authors how to write stories*. This formula (described in detail in Part II) divides stories into five sections. AGAST attempts to model this formula by creating stories in the form of five inter-related sections.

AGAST is flexible because it "writes" stories in two stages. The first stage creates a story tree, where every action that happens is stored. As the tree in generated, the internal representation of the physical world (including locations, objects, and characters) is affected, which in turn affects the progress of the story.

A straight-forward depth-first traversal of the story tree produces a "chronological" account of the story. The second part, the text generator, walks the tree this way and thus tells the story. This part is extremely simple now; it just writes a sentence for each and every action, telling the story in excruciating detail. But since the story structure is unaffected by the telling, a different text

----
* i.e. to capture what is essential in a story as opposed to a random collection of sentences, or some other form of prosaic writing such as a newspaper article or master's thesis

generator could easily be used before, after, or instead of the one used now. Sequences of events could be summarized to different levels as needed, events could be told in varying orders, or two stories could be meshed together.

The fact that the story exists as a tree after its generation means also that actions can be undone and the story can take a new direction in the retelling. AGAST uses this feature to handle stories that end in story-telling failure. Like a human writer, AGAST can "change its mind" and rewrite the story to end successfully. This also means that AGAST doesn’t need to plan every detail of the plot ahead of time. It can randomly generate complications for the plot, handle them using using any sort of decision making process, and know that if it paints itself into a corner, it can either undo the actions that got it into trouble, or change the situation so that the characters can successfully handle the problem.

Related Work

A. James Meehan's "TALE-SPIN"

"Tale-Spin" [Meehan 76] seems to be the grandparent of computer story writing--at least, everyone who does work in the field must refer to James Meehan's ground-breaking work.

Like Tale-Spin, AGAST has a physical world, with objects and locations. Both have actions, although AGAST's actions are arranged in a writerly (GIVE, PICK-UP) rather than a formal (PTRANS, MTRANS) fashion.

Tale-Spin's stories are generated by goal stacks--the original goal puts other goals on the stack, the achievement of all of which completes the original goal. AGAST's stories have this feature as well. However, like AGAST actions, AGAST goals are arranged in a writerly fashion, each broken down into story parts. In addition there is the possibility of having multiple goals, all being solved simultaniously.

One example of this is when a character is looking for two different objects. First one object is sought, and when it is found, the other is sought. However, if the second object is run across in the search for the first, it is picked up and the search for it is never initiated. Although it isn't yet implemented in AGAST, characters can have other character's goals as subgoals, thus helping friends achieve their major goals.

Also, in storing actions and their side-effects as they occur, AGAST allows story revision and backpatching, which aren't conceivable in Tale-Spin.

Tale-Spin does have some level of social interaction, which AGAST is at present totally missing.

B. Natalie Dehn's thesis

Natalie Dehn [Dehn 81] makes the point that in writing a story, authors have a goal: to write an interesting story. Her project concentrates on author intentionality. AGAST attempts to emulate this goal with the plot formula that drives the story, and with backpatching that "saves" the story when it plots its way into a dead end.

However, there is also the point that characters must have goals. If they start without a specific goal, they are quickly given one, from simply staying alive to saving the universe. Dehn points this out (but not in terms of character goals) when she mentions justifying the situation a character finds him or herself in. A story goes wherever the author intends, but it won't be a very good story if the characters seem to be acting only on the author's whim. They should be following their own goals; their actions should make sense to them, not just to "The Story," of which characters generally aren't aware, anyway. AGAST attempts to combine the internal logic of goal-driven behavior of Tale-Spin with the author-intention-driven stories that Dehn promotes.

C. Michael Dyer's "BORIS"

Michael Dyer's work [Dyer 81] is more on story understanding than on story generation. Dyer's BORIS attempts to understand stories not only by general semantic, grammatical and lexical knowledge but by discerning the context that the story creates. AGAST creates and stores its context, but so far makes only a limited use of it. One example in which AGAST uses the context of an event is when an accident occurs (a character is injured--they trip, or some such accident). If the character is just travelling or exploring, they can cure themselves (but only if they're carrying a medikit). However, if they are fighting or escaping, they can't take the time to do anything about the injury.

While it would surely be interesting to have BORIS read in AGAST stories and answer questions about them, it would be more interesting to have a BORIS-like program enhance the context that AGAST builds. A memory of past events would allow characters to "learn" and would make social interaction easier to simulate. For instance, suppose Frank killed Libby's cat. When Libby next meets Frank, the past event might make her want to get revenge on Frank, and thus would influence what she did during the meeting. As in Tale-Spin, she would know Frank was not to be trusted--but she would conclude it rather than knowing it from the start. From the examples in Dyer's paper, it would seem possible to use such a system to determine characters' attitudes toward other characters and their current emotional states.

D. Eduard H. Hovy's "PAULINE"

The actual text of AGAST stories is generated very simply--every object in a story tree knows how to print a description of the action it represents. This produces very lengthy, boring text (see sample stories).

Eduard H. Hovy [Hovy 87] discusses a much better text generation model. His program, PAULINE, groups related actions together and summarizes them, specifically mentioning only the "high points" of the event. PAULINE interprets the actions, draws conclusions, and adds them to its knowledge of the event. PAULINE can also "shade" what it tells, adding evocative words that can slant the meaning of the text, although the event is still accurately portrayed.

These abilities would greatly enhance the "story-ness" of AGAST's stories. Instead of:

Libby swung her sword at the giant centipede, injuring its leg.
The giant centipede bit Libby, injuring her arm.

...and so on, each exchanging many blows and ending with:

Libby swung her sword at the giant centipede, injuring its head. Its head was severed and dropped to the floor. The giant centipede was killed.

A program such as PAULINE might be able to produce more writerly text:

Libby drew her sword as the giant centipede attacked. She slashed at the slavering creature as it bit at her. Howling with rage, the giant centipede sank its mandibles into Libby's left arm. Libby raised her sword and with a cry of desperation cut off the centipede's head.

Since AGAST's actions are already grouped and catalogued (a series of "injure" actions that constitute a "fight" are stored in a slot of a "fight" event), summarizing events and choosing weighted words that fit the situation (desperate, rage) and the characters in it ("slavering", since giant centipedes are defined as non-intelligent animals) should be relatively easy to do.

E. Michael Lebowitz's "UNIVERSE"

Michael Lebowitz's UNIVERSE program generates plots for soap-opera-like stories. This is different from most other work in the field in that stories in UNIVERSE are deliberately constructed not to end, but to have continuing characters moving from mishap to mishap. Below we examine two versions of UNIVERSE which have appeared in the literature.

[Lebowitz 83]

While AGAST is action-heavy, Lebowitz is primarily concerned with character consistency and development. Past events affect the personality (and thus actions taken) of characters. Like UNIVERSE, AGAST creates important characters--the protagonist, the antagonist, anyone directly involved in the main goal--before the story starts. Unimportant characters--attacking monsters, for instance--are created on the fly.

AGAST reaches for this ideal with the Background section of the story providing the motivation for the story. However, UNIVERSE goes much further, with each character carrying around a changing history. Like Dyer's BORIS, a UNIVERSE-like program could significantly help the character development of AGAST's stories.

UNIVERSE also keeps track of character relationships in a more consistent way than AGAST. AGAST's characters can be related to one another (e.g. Libby is Frank's mother, and Frank is Libby's son), but if Frank is Stella's brother, all Libby knows is that she is somehow related to Stella. Relationships that change with time, such as marriage, exist, but are assumed permanent, with no history of past divorces or whatever.

[Lebowitz 87]

Using goal precedence and mutual-achievement critieria for goal selection, UNIVERSE nicely manages the interweaving of plots that is important for a complex, interesting story. AGAST currently has only one form of subplot implemented, the substory. Here the main goal is temporarily suspended while a subgoal (with a "subprotagonist") is "written" in a "meanwhile, back at the ranch..." type of story. The conclusion of the subgoal, usually with the subprotagonist and the protagonist joined as companions, then allows the completion of the major goal. The nesting of substories, however, can produce quite complex stories (Libby rescues Fred; they both rescue John; all three join with Natasha to continue the search for the Lost Ark of the Covenant...).

The "churning" of plots that UNIVERSE uses as one of its author goals is somewhat emulated by the introduction of obstacles and problems into the path of the protagonist. Plans have particular problems associated with them, and goals (which determine the type of story, such as the "quest") can also have special problems associated with them, especially at the climax of the story (which is a concept UNIVERSE doesn’t have, since it writes "slice of life" narratives).

F. Schank and Abelson’s scripts

Schank and Abelson [Schank and Abelson 77] discuss many of the methods used in story writing programs. One important idea is that of scripts--an outline of how to behave in particular situations. They allow both understanding and generation of simple stories involving frequently done events, such as eating in restaurants or taking the bus.

Actions in the AGAST story tree are grouped and stored in PLAN objects. Many plans are similar to scripts in that they generate a restricted series of actions that constitute a type of event. For instance, if the event is a FIGHT between two parties: first, a character (randomly chosen from the first party) injures one (randomly chosen from the second party), then a return injury is done. These actions are repeated until one of the parties has no one left who can continue fighting.