Assisting Writing Through Step-by-Step Story Generation

Carlos León
Dpto. de Ingeniería del Software e Inteligencia Artificial
U. Complutense de Madrid
cleon@fdi.ucm.es

Pablo Gervás
Instituto de Tecnología del Conocimiento
U. Complutense de Madrid
pgervas@sip.ucm.es

SAC 2011 2011, Nov 3, 2011, Atlanta, USA.
ACM.

Abstract
Most automatic story generation systems do not explicitly contemplate assisted writing by design. They are usually focused on a pure generation in which the user expects a full story that must show a good quality according to some criteria. While this is a challenging task by itself and it helps to advance in the field of story generation, the resulting systems are not directly usable in general. In this paper we present a design methodology for story generation systems that is based on a step-by-step generation. The proposed methodology proposes building a system that explores a generation space in which the starting story is the partial story written by the user and offers a set of solutions every time the human writer asks for it. In this way, many current story generation techniques can be used in assisted writing without having to create new paradigms.

Keywords
Story generation, assisted generation, artificial intelligence

ACM Classification Keywords
narrative, story, computer supported/ mediated creativity, generative

General Terms

Algorithms, design.

Introduction

Storytelling is one of the most important forms of communication. As such, great effort is put everywhere and every day to invent, develop and create new stories. This makes it very useful to build a set of tools to ease this time-consuming process.

There exist many methodologies for writing stories by hand, basically oriented to train storytellers. These methods basically give more or less generic rules to engage an audience or facilitate the understanding of the message being told. However, while these influence or improve the creativity of the generated narrative content, they fully rely on the effort of the storyteller.

To reduce the amount of effort that is required to create a new story, automatic story generation systems have been created since the 70s. These Artificial Intelligence programs create full stories from scratch by following many different approaches and with very varying results.

While these story generation systems can produce new, more or less original content, they are neither used nor designed for assisted writing. The current state of the art is perhaps slightly away from building fully usable systems, but it makes sense to try to take one step further by putting together what has been learnt so far. While this inherent difficulty makes it very effort consuming to implement a story generation system able to generate many full, good quality stories, automatic systems can be used, at least, as a useful tool for partially suggesting plausible additions to a story.

In this paper we propose a design methodology for story generation systems in which the generation engine is able to receive a partial story and outputs a set of possible additions to it, and not necessarily a full story. In this way, partial, step-by-step generation can be used to assist literary writing without having to change the underlying Artificial Intelligence paradigm used when building story generation systems.

Other Story Generation Systems

Several story generation systems have been created so far, but the corresponding research has not put the focus on the design of the system but on generative issues instead and their relation with assisted writing.

TALE-SPIN tackles story generation as a problem solving process [1]. Characters in TALE-SPIN have problems that they must solve, and during the solving process the story emerges. This design proposes a basic core engine reasonably independent from the set of actions that can be performed. The main limitation of this system is the simplistic assumption that narratives can be only represented by problem solving schemas. The planning nature of TALE-SPIN makes it a suitable system for step-by-step generation.

Grammar-based story generation systems propose a top-down approach for the design of the generation systems [2,3]. Grammar-based generation systems do not scale well because the dependency of the added knowledge with the rest of the knowledge base is not well managed. Planning (which confers the characters a form of believability) is based on templates and is not

powerful enough. In general grammar-based generation assumes a full story and therefore does not fit partial generation well.

FABULIST focuses on implementing character believability and well-caused plots [4]. FABULIST generates stories by applying a planning algorithm that applies a set of actions that define the domain. This research applies both the idea of generic planning to build stories and the inclusion of a model of character intentionality. This kind of generation model fits well partial generation.

MEXICA tries to implement Sharples' model of engagement and reflection as the human process for creativity, in particular in story generation [5, 6]. Since MEXICA is based on sequential generation stages, it could be adapted for step-by-step generation.

BRUTUS is another example of a story generation system that focuses on the study of creativity [7]. The set of output stories, while small, includes instances of good quality. BRUTUS' engine consists on a large knowledge base that encodes a significant amount of common sense knowledge and narrative. Being a grammar based approach, BRUTUS would not be, in principle, a good candidate for partial generation.

Another knowledge-intensive system is presented by León and Gervás in a story generation system based on evaluation and selection of stories [8]. The state space search implemented is this system could be easily adapted for step-by-step generation.

There are also other story generation systems including other original concepts [9,10]. While these are influential in general, only the most relevant ones regarding this research have been described.

The rest of this paper describes an effort to abstract many of the previously presented concepts and ideas in a general design that tries to exemplify how the process of writing a story generation system focused on assisted creation can be developed.

Step-by-step Story Generation Engine

This section describes the core engine of a design that can help to define an automated system for assisted writing. The definition of this approach makes several assumptions in order to allow for a restricted description of the engine. These assumptions can be made because only narrative generation is contemplated. Therefore, nothing is claimed about the generalization of this methodology for knowledge-based Artificial Intelligence generation systems in general or other assistive technologies.

The main assumptions that are made for this proposed computational design are:

- Narrations take place during a certain interval and the specific details of what happened before the story and what will happen after it are not important. That is, the narration itself defines what the computational system must manage.

- The audience completes the missing content when a story is communicated. This implies that only communicative success is intended and, as most Artificial Intelligence systems assume, there is no need for emulating human

behavior if the output is useful. Creativity can arise, therefore, just as a phenomenon resulting from the interaction between the generation and the audience.

This list does not try to define narrative or computational generation of creative artifacts. Instead, it includes a set of assumptions that are considered to be valid in order to simplify the design of story generation systems, especially for assisted generation in which the final output does not have to be final but helpful.

The rest of this section describes a high level abstract formalism of stories and the proposed methodology by defining the set of properties of the basic generation engine. Some definitions and formalisms are equivalent or easily translatable to other representations or models. The ones presented here correspond to the prototype implementation of the methodology.

Formalism for stories

Stories can be represented as ordered lists of event descriptions. An event description contains an action and a set of parameters. In this way events in a story are represented as structured data that is atomic in the sense that each one of these structures describes a specific action in time and cannot be sub-divided in smaller parts.

Narrative plots can then be easily represented with this formalism, as it can be seen in many other approaches to story generation which have used a similar representation [2,4,8]. Lists of event descriptions with time information are usually enough to represent most simple plots. Since no text output is addressed in this paper, this formalism is enough.

Core Generation Engine

The proposed core engine performs story generation as a generic state space search problem, although any generative methodology capable of issuing one addition to a story at a time can be used as the core engine in the current proposal.

The story itself represents the state, and it is updated as the story is completed. According to this, the final objective is to generate a story that fulfils the user requirements, that is, to reach a state in which all user's constraints are satisfied. It can be assumed that the user's constraints are a set of rules about the objective story.

At some stage, then, the engine is in a certain state. This state can be initial (empty story) or partial. In any case, the generation changes its state to a new one that is not necessarily final. This state has been produced by some set of implicit or explicit rules (this part has many possible implementations and its definition is beyond the scope of this paper). This part of the abstract system that is being defined is the crucial part of the model. The key idea is that the generation system should be able to generate and issue each event description instead of trying to create a whole story. Some of the systems previously reviewed could be used in these terms.

This model implies that any partial story can be used as input for the generation process. An empty story would create a story from scratch, but also a partially written story could be used to launch the generation. In this

case, the story generator would complete what an external source (maybe a human writer) had previously written. In this way, the proposed methodology can be the core of a story generation assisting tool.

Assisted Generation

A system like the presented one can be used as the core engine of an assisted generation tool. As previously introduced, the generation engine assumes a computational, structured representation of the story in the form of an ordered sequence of events. If the assumption that either a bidirectional text-to-event translation tool is available or the story is encoded in a non-textual formalism is made, the generation engine can be provided with a computational definition of the partial story written by a human author.

At that point, the human author might be unable to come up with any good addition to the story, so she or he can ask for a suggestion trough some hypothetical interface. Once the user does that, the story generation system would perform a new generation step.

This step would make the core generation engine read the partial story (and probably encode it in a structured way). This would represent the initial state of the search. After having processed the input as shown in the previous section, a set of candidate additions would be presented to the user, who would then choose one of them (or maybe none) and then go on with the writing.

Several improvements to this model make sense. For instance, the list of candidate additions could be automatically filtered with some set of user-defined constraints. The candidate additions could be composed by elements representing more than one event without having to rewrite the engine because all the user interface has to do is to perform two cycles of generation.

The proposed design methodology can then be considered as a generalization of story generation systems in which every generation step is offered to the user.

Discussion

Assisted narrative creativity can be improved by a tool that performs automatic generation. If a user has the ability of getting several possible completions from a previous plot that she might have written, she has more material to work with. Maybe one of the stories is the perfect candidate, or maybe a story inspires the writing process. The main benefit of doing this with the help of an automatic system is that the required effort by the user can be greatly lowered.

The proposed schema has been designed to ease the solution of the problem and to partially make it simpler to build narrative generation systems. The main assumption behind this research is that automatic story generation is helpful as an assisted method. In the authors' opinion, we are still far from being able to replace human criteria and knowledge management capabilities. Therefore, the proposed system, as exemplified in the case study, is only a step forward as an assisting tool.

Authors do not claim, however, that the whole problem of creative story generation system can be solved by automatic generation. However, this proposed system

is defined in such a way that creativity can not only be assisted by having a new story generated, but also by the fact that the domain definition is made simpler than in other approaches. In this way, narrative creativity can be enhanced by letting the writer add new content to the system.

Not every problem of story generation has been solved. Knowledge management is still a problem, and efficiency is a very important issue, especially in assisting environments. While these and other aspects are usually studied and tackled in pure story generation, they must not be left out when designing or studying a generation-based assisting.

The current model has a limitation: it only considers additions to the current story. A complete assisting system should be able, however, to suggest other kinds of modifications like changes in the text or deletions. While this has not been explicitly addressed in the paper, it is assumed that the proposed mechanism would be capable of performing these operations after some modifications. For instance, deletions could be suggested by the system by performing generation and outputting a story that is similar to the current one except for one event. This, however, must be further developed as part of the future work.

Conclusion

Automated story generation systems can reduce the amount of work in storytelling when used as an assisting tool, and this can help to spend the effort on the creative part. Since developing this kind of tools requires a complex handling of Artificial Intelligence techniques, this paper suggests a methodology in order to ease the creation of story generators. Several assumptions that can be made because of the constrained domain (narrative) help to restrict the constraints of the problem so that a simpler definition is possible.

More research has to be carried out. Given that evaluating this kind of systems generally implies some form of experimentation, the theoretical aspects, while important, can only be validated through a robust implementation and empirical evidence. An implementation is planned as part of the future work.

While it is considered that the methodology is promising, nothing definitive can yet be claimed. It is expected that the practical evolution of the system can help to identify new patterns and offer new solutions. It is assumed that creativity in storytelling can be more easily assisted by these means.

One of the short-term objectives, then, is to complete an implementation of an assisted writing environment. Such a system will allow for real experimentation that will give non-speculative evidence of the proposed assumptions and conclusions. The authors expect that, after having implemented the system, assisted writing based on story generation can become a reality, at least as a proof of concept until story generation is able to produce high quality stories.

Acknowledgements

Research funded by Spanish Ministry of Science and Innovation’s TIN2009-14659-C03-01 and GR35/10-A (group 921332).

References

[1] Meehan, J. (1976). The Metanovel: Writing Stories by Computer. PhD thesis, Yale University.

[2] Lang, R. R. (1999). A declarative model for simple narratives. In Proceedings of the AAAI Fall Symposium on Narrative Intelligence, pages 134–141. AAAI Press.

[3] Rumelhart, D. E. (1975). Notes on a schema for stories. Representation and Understanding: Studies in Cognitive Science, pages 211–236.

[4] Riedl, M. (2004). Narrative Planning: Balancing Plot and Character. PhD thesis, Department of Computer Science, North Carolina State University.

[5] Sharples, M. (1996). An account of writing as creative design. The Science of Writing.

[6] Pérez y Pérez, R. (1999). MEXICA: A Computer Model of Creativity in Writing. PhD thesis, The University of Sussex.

[7] Bringsjord, S. and Ferrucci, D. (1999). Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a StoryTelling Machine. Lawrence Erlbaum Associates, Hillsdale, NJ.

[8] León, C., Gervás, P. Towards a Black Box Approximation to Human Processing of Narratives based on Heuristics over Surface Form. AAAI 2010 Fall Symposium on Computational Models of Narrative. Arlington, Virginia, pp. 31-38.

[9] Lebowitz, M. (1985). Storytelling as Planning and Learning. Poetics, 14:483–502.

[10] Dehn, N. (1981). Story Generation After Tale-Spin. In Proceedings of the International Joint Conference on Artificial Intelligence, pages 16–18.