Creative Language Generation in a Society of Engagement and Reflection

George A. Wright and **Matthew Purver**

Cognitive Science Research Group
School of Electronic Engineering and Computer Science
Queen Mary University of London
{george.a.wright, m.purver}@qmul.ac.uk

Abstract

Many existing models of narrative and language generation use rigid sequences of steps which are cognitively implausible and limit creativity. Iterative models based on Sharples' *cycle of engagement and reflection* improve on this by incorporating self-evaluation but still have a rigid arrangement of parts. This paper outlines how a multi-agent approach could be used to break apart the cycle into a more fluid *society of engagement and reflection*, whose constituent agents interact with one another to produce a text. Our approach is to work in a simple domain in order to focus on the underlying processes, and to avoid the ELIZA effect during evaluation.

Introduction

Narrative is how humans make sense of the world. A model of narrative generation is thus an important strand in the development of intelligent and creative machines. But, much AI and CC work on narrative generation focuses on efficient yet rigid generation of textual summaries and/or the generation of stories and scenarios in an interesting, literary domain. There tends to be less focus on the processes that take place in a human mind during the creation of a narrative text. This paper outlines first steps towards a model based on the interactions of micro-agents which should approximate theories of cognition such as Minsky's (1986) *Society of Mind*.

The Question of Architecture

Many models of narrative and language generation use a fixed sequence of discrete steps. This is best exemplified by the data-to-text pipelines used for summarizing structured data, although neural architectures also tend to be unidirectional and run in a fixed order. The pipeline approach has been applied to many tasks, including, recently, to the description of election results (Leppänen et al. 2017). Reiter (2007) divides the data-to-text pipeline into four stages:

Signal Analysis A search for patterns in the data.

Data Interpretation Identification of "messages" from the patterns and relations between messages.

Document Planning Selection of messages and arrangement into a rhetorical structure.

Microplanning and Realization Generation of natural language text.

The stages of the pipeline explain the processes a human goes through when describing data. Indeed the work of Reiter and his colleagues is (at least in part) inspired by observations of humans (Yu et al. 2006), but the fixed, uni-directional arrangement of the processes is not realistic.

Greater realism is offered by Sharples' (1998) *cycle of engagement and reflection*, partly implemented in MEX-ICA (Pérez y Pérez and Sharples 2001), which is divided into stages slightly analogous to those in Reiter's pipeline:

Contemplate Form ideas ($\approx$ Signal Analysis + Data Interpretation).

Specify Select and organize ideas ($\approx$ Document Planning).

Generate Produce text ($\approx$ Microplanning and Realization).

Interpret Review and interpret generated text.

The *generate* stage belongs to *engagement*, the others to *reflection*. The cycle restarts after interpretation, allowing for a consequent re-working of the text. This is more in tune with evidence from psychology and neuroscience that language production and comprehension are intertwined (Pickering and Garrod 2013). But large, self-encapsulated modules in fixed positions cannot fully account for this intertwining, nor for the fluidity and spontaneity we expect from what Fauconnier and Turner (2002, p321) term the "bubble chamber of the brain". This is the case with many models, even those using sophisticated techniques for each module such as neural networks (Fan, Lewis, and Dauphin 2019) or genetic algorithms (McIntyre and Lapata 2010).

The FARG Approach More fluidity and spontaneity occurs in the models of analogy making and creativity by Hofstadter and his Fluid Analogies Research Group which consist of thousands of small agents called *codelets* that gradually build (and sometimes destroy) structures in a workspace (Hofstadter and FARG 1995).

One of their earlier models is Copycat, which solves analogy problems of the form "if *ABC* goes to *ABD*, what does *XYZ* go to?" (Mitchell 1993). Similar methods have been applied to other areas such as music understanding (Nichols 2012) and typeface design (Rehling and Hofstadter 2004).

Copycat tends to produce more sensible solutions to problems, but when faced with an unusual situation can come up with less obvious solutions (such as *WYZ* to the above

problem). Hofstadter compares this to the way people resist "nonstandard ways of looking at situations" unless a change in circumstances warrants it (Hofstadter and FARG 1995, p240). The usual answer to an analogy problem like the one above would be to replace the last letter with its successor in the alphabet, only in the case of *XYZ* that is not possible, so a more outlandish approach is taken involving a reversal.

The Copycat architecture has three main components:

The Workspace where an initial problem is perceived and structures are built by codelets to represent groupings and analogical mappings. The workspace has a *temperature* indicating the coherence of its structures.

The Slipnet a semantic network whose nodes spread activation and slip towards and away from one another according to the current context. Active nodes send *top-down* codelets to seek instances of their concept.

The Coderack where codelets are selected stochastically and according to their urgency. If the workspace has low coherence, selection is more random, and more open-minded *bottom-up* codelets can explore alternative paths.

In general, *top-down* codelets become more dominant over time as the temperature (non-monotonically) decreases and a single path to a solution is chosen. It is possible that a chosen path will result in a snag — in which case the temperature will increase, offending structures will be destroyed, and alternative pathways will be considered (Mitchell 1993).

Unlike the frameworks for language and narrative generation discussed above, FARGitecture does not involve a central authority directing the model through stages in a sequence: control is distributed between codelets and slipnet nodes. When more bottom-up codelets are running, the system is in a relative state of reflection (contemplating new structures and reviewing existing ones), while when more top-down codelets are running, the system is in a relative state of engagement (pursuing a particular path towards a solution). FARGitecture therefore enables a fuzzy alternation between engagement and reflection.

Copycat's lack of central control, tendency to vary its behaviour due to stochasticity, and ability to pursue stranger solutions when circumstances allow make its architecture more cognitively plausible than other more rigid models.

The Question of Domain

This paper outlines how ideas developed by Hofstadter and FARG (1995) could be applied to narrative generation. Their approach is to work in micro-domains so that evaluation must focus on the decisions a program makes while exploring its search space, not on any meaning inherent to the space. This is a different approach from most work in creative language generation which tends to cite Meehan (1976) as the earliest work in the field while overlooking the more modest (yet more impressive) work of Davey (1974). Whereas Meehan's TALE-SPIN generates stories about animals living in a forest, Davey's PROTEUS narrates games of tic-tac-toe. PROTEUS' subject matter is boring but its use of features such as co-reference and conjunctions produces highly readable pieces of text. TALE-

SPIN, on the other hand, outputs stories as lists of self-contained pseudo-English sentences which are easy to understand but aesthetically displeasing. Work on creative language generation tends to deal in overtly literary domains. But, all language is creative: even a tic-tac-toe commentator has to make decisions about how to structure a text; how terse or detailed to be; and what words to use where.

At this early stage in the path towards creative machines, research should avoid complex, literary domains which give the impression of creativity where there is none, and first see how decisions can be made in a simpler domain of discourse. This will prevent evaluators from succumbing to the ELIZA effect — jumping to the conclusion that a machine has achieved human levels of intelligence when it really only relies on a few simple tricks. Veale (2017) shows that, when using the same method to build plot skeletons, giving characters the names of celebrities results in higher ratings for dimensions including *imagination* and *drama* than when using generic animal characters. Readers cannot help but find meaning in a text which the artificial author is oblivious to.

Following FARG and Davey, this paper outlines a proposed architecture for narrative generation intended for testing on mini-domains such as weather and board games.

Describing a day's weather forecast involves recognizing entities such as storms and patches of warm or cold weather; tracking their movements and changes; and weaving together these threads into one linear piece of text. Certain aspects of narrative are lacking from this domain: for example, there is no need to account for characters or their motivations. But describing the weather does require many mechanisms fundamental to narration: formulating a narrative of the weather requires the ability to select interesting pieces of information; discard other pieces; find appropriate names for the entities that have been recognized; and to find a good structure for the text. There are many non-trivial issues to tackle — even in this simple domain.

Board game narration is a domain that could provide some of the other ingredients of narrative: there are characters with goals and plans (the players), and there is space for imagined counterfactuals. In some ways board games are simpler than the weather: entities in checkers and chess are discrete whereas weather patterns have fuzzy boundaries. Board games also have a clearer beginning and end.

Ultimately, an architecture that could handle both of these domains would be a good candidate for a general model of humans' storytelling capacity. This paper focuses, for the most part, on the domain of weather.

A Society of Engagement and Reflection

In this (yet unimplemented) architecture everything is done by codelets, including: data interpretation; arrangement of the text; language realization; evaluation of structures; and destruction of those that are no longer wanted. These tasks correspond to the modules in pipeline and cyclic architectures discussed above, but while most models perform these functions in a strict order, in this society model the tasks are broken down into small units of work which can be carried out whenever appropriate. A codelet runs not according to its position in a line-up, but due to competing data-driven

bottom-up pressures and conceptual and aesthetic *top-down* pressures.

Each codelet can be classed as either *bottom-up* or *top-down*. Bottom-up codelets are more open-minded, looking for anything of interest, whereas *top-down* codelets are more single-minded, looking for instances of a specific concept.

Data Labeling and Grouping Codelets Bottom-up data interpreting codelets access raw data in the workspace and determine the best concept with which to label it. For example, in the weather domain, a location with a temperature of 25°C may be labeled HOT. This leads to the HOT semantic network node receiving a boost in activation. Once fully activated, this node sends out top-down codelets to look for other locations that can be labeled as HOT. After a while, many of the same labels begin to appear in one region of the map and grouping codelets, recognizing the similarity, divide the map into regions corresponding to weather type.

These codelets perform a similar role to a convolutional kernel in a neural network, indeed they could each be implemented as a neural or other machine learning classifier. The benefit of using individual codelets which are run according to the urgency determined by activations in a semantic network, instead of having fixed layers in a neural network, is that they are not necessarily run unless the combination of context and top-down desires deems it necessary. For example, having recognized a pattern of interest in the north of a map, the NORTH node in the semantic network may spread activation to the SOUTH node to encourage a search for a pattern which summarizes the south. This architecture of interacting codelets allows for higher-level relational processing to be followed by a reversion to lower-level raw-data processing similar to how Yu et al (2006) found experts switch between more coarse and more detailed views when analyzing data to get "details-on-demand". Feed-forward neural architectures and traditional pipeline architectures, on the other hand, rely on all of the data interpretation that could possibly be relevant having been done at an early stage.

Language Generation Codelets Several codelets perform the task of microplanning and realization.

Phrase codelets recognize a structure that can be transformed into a phrase. E.g. *rainy → It will be rainy.*

Connective codelets recognize two phrases which can be joined. E.g. *It will be rainy. It will be cold. → It will be rainy and it will be cold.*

Deletion codelets remove unnecessary parts of a phrase once it has been connected. E.g. *It will be rainy and it will be cold. → It will be rainy and cold.*

Ordering codelets order two or more phrases or sentences, such as in a general-to-specific order or along a dimension of a conceptual space. E.g. *It will be warm in the midlands. It will be hot in the south. It will be cold in the north → It will be cold in the north. It will be warm in the midlands. It will be hot in the south.*

Phrase codelets essentially apply templates. But, the aim is to limit the size of templates and allow for them to be combined, re-ordered and re-structured in order to limit repetitiveness. This is similar to the approach taken by Leppänen

et al (2017), but this architecture should allow for more diverse realizations. For example, there may be different ways to order phrases according to the most salient concepts in the context; and there may be different ways to connect phrases according to how ordinary their co-occurence is: *hot but rainy* makes sense; *cold but rainy* does not (at least from a British perspective). The exact realisation that the architecture chooses will in part depend on its stochasticity and it will not be expected to re-produce the same text if run again.

Other codelets are also required, such as those that arrange rhetorical structure and those that pick which information to include in the text.

A Hypothetical Example

Figure 1 is an example of a map of the weather at a point in time for the model to describe (more realistically, it should handle a sequence of maps in order to qualify as narrative). This map has four *channels*: weather type, wind (direction and speed in kph), temperature (in centigrade) and percentage probability of precipitation. Below is an example of a textual forecast it might generate.

> *It will be cloudy in the north with a high chance of rain and furthermore snow in the very north. There will be dry weather in the rest of the country but there may be pockets of rain in the south. It will be sunny in western and central areas but temperatures will be mild while it will be cloudy but warm in the southeast.*
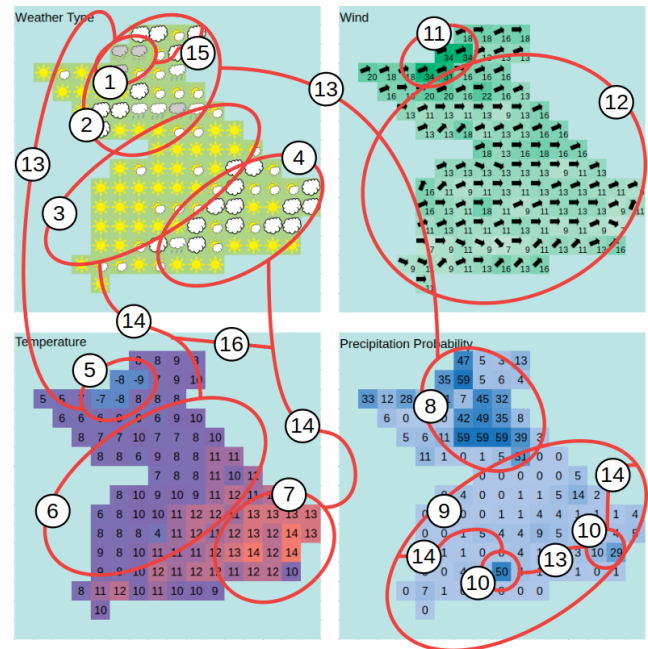

Figure 1: A four-channel map of weather in Britain with groups and relations. **1-12**: Regions of similar weather; **13**: AND relations; **14**: BUT relations; **15**: A FURTHERMORE relation; **16**: A second-order AND relation. (Data from the Met Office).

At the start of the program's run bottom-up codelets search for the types of weather present on the map. Codelets also group them into regions. The ellipses in figure 1 indicate approximate regions that might be recognized.

Codelets then find relations between regions. Certain regions are recognized as being to some extent the same, for example regions 2 and 8 in the north of the country. The north's cloudy weather and high chance of rain are ordinarily co-occurring types of weather thus are connected with AND. Meanwhile the south's cloudiness and warmth are less typical so are connected by BUT. When a sub-region has a more extreme kind of weather than its parent region, for example the snow in a small part of the north, a FURTHERMORE relation is used. When a temporal sequence of events is being described, yet more relations can be recognized, such as THEN and THEREFORE. Higher-order relations are also possible: 16 shows an AND connecting two parallel BUTs.

Codelets use weather, location, and relation labels to begin forming phrases. Certain labels depend only on local concepts such as "the north", while others such as "the rest of the country" are context-sensitive.

Arrangement of the text also depends on linguistic context. For example, the sentence describing *the rest of the country* must come after the sentence describing *the north* in order for *the rest* to make sense. The sentence comparing the western and central areas and the southeast ought to come last since it is an elaboration of the sentence describing *the rest of the country*.

Codelets must recognize the importance of context and discourse relations as they arrange the final text.

Open Questions

Many questions need to be answered in order to get this architecture working: what conceptual knowledge will the model require? Can the model be applied to board game narration and beyond? How much of the workspace context must each codelet be aware of? How will the model handle complex situations where concepts have varying relevance in different places?

This last issue, French (1995) describes as the "problem of single nodes with multiple activations". It was a major problem in his (FARGitecture based) model of analogy making between objects on a dinner table, and required a hierarchy of different contexts corresponding to different patterns of activation in the semantic network. It is likely to be an even larger problem in narrative formation, which can involve summarizing even more situations than when making a single analogy.

Conclusion

There remain issues to be resolved in applying this style of architecture to narrative generation, but its potential for flexibility makes it an attractive line of research. Work so far has centred around the mundane domain of weather so that focus can be placed on the most fundamental issues involved in narrative and language. Future work should move into richer domains such as board game narration in order to better test the generality of the approach.

References

Davey, A. 1974. The formalisation of discourse production. PhD Thesis, University of Edinburgh.

Fan, A.; Lewis, M.; and Dauphin, Y. 2019. Strategies for structuring story generation. In *Proc of the 57th Annual Meeting of the ACL*, 2650–2660.

Fauconnier, G., and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.

French, R. M. 1995. *The Subtlety of Sameness: A Theory and Computer Model of Analogy-Making*. The MIT Press.

Hofstadter, D., and FARG. 1995. *Fluid Concepts and Creative Analogies*. Basic Books.

Leppänen, L.; Munezero, M.; Granroth-Wilding, M.; and Toivonen, H. 2017. Data-driven news generation for automated journalism. In *Proc of the 10th INLG*, 188–197.

McIntyre, N., and Lapata, M. 2010. Plot induction and evolutionary search for story generation. In *Proc of the 48th Annual Meeting of the ACL*, 1562–1572.

Meehan, J. R. 1976. The metanovel: Writing stories by computer. PhD Thesis, Yale University.

Minsky, M. 1986. *The Society of Mind*. Picador.

Mitchell, M. 1993. *Analogy-Making as Perception: A Computer Model*. The MIT Press.

Nichols, E. P. 2012. Musicat: A computer model of musical listening and analogy-making. PhD Thesis, Indiana University.

Pickering, M. J., and Garrod, S. 2013. An integrated theory of language production and comprehension. *Behavioural and Brain Sciences* 36:329–392.

Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *JETAI* 13.

Rehling, J., and Hofstadter, D. 2004. Letter spirit: A model of visual creativity. In *Proc of the 6th ICCM*, 249–254.

Reiter, E. 2007. An architecture for data-to-text systems. In *Proc of the 11th ENLG*, 97–104.

Sharples, M. 1998. *How We Write: Writing as Creative Design*. Routledge.

Veale, T. 2017. Déjà vu all over again: On the creative value of familiar elements in the telling of original tales. In *Proc of the 8th ICCC*, 245–252.

Yu, J.; Reiter, E.; Hunter, J.; and Mellish, C. 2006. Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering* 13:25–49.