An Evaluation of the Impact of Constraints on the Perceived Creativity of Narrative Generating Software

Lewis Mckeown

School of Computing
University of Kent
Canterbury
lam54@kent.ac.uk

Abstract

There are a variety of ways to understand the computational generation of narratives. This project attempts to categorise them as a spectrum which arises from the application of constraint. To assess the validity of the spectrum, the literature surrounding narrative generation systems is introduced, as are ways of evaluating their creativity. This provides the groundwork for an investigation into the impact of constraints on the perceived creativity of the output of narrative generating systems. The investigation aims to understand what level of constraint application results in the most creative output. To achieve this, software is written that generates short stories, using adjustable levels of constraint meant to reflect those utilised by software at different positions on the spectrum. The creativity of the output is then assessed by human evaluators. The results are promising and show a clear variation of response based on the level of constraint imposed on the narrative generation process. This supports the assessment of narrative generation software in terms of a spectrum of constraint application. The results show a sweet spot for maximal creativity closer to the less constrained end of the spectrum, which demonstrates the potential for more creative software by the relaxing of constraints. If a system strays too close to randomness however, the perceived creativity will be heavily penalised. In contrast the strictest application of constraint showed the second highest level of creativity. This is a fine line, as too relaxed or too moderate applications of constraint will result in much lower creativity ratings.

Introduction

Computers and Creativity

Computers and the software running on them are generally seen as supportive tools for human intellectual endeavours, or as a means of automating tedious tasks. The idea of computers making something creative, independently and without human interaction, although not new, has expanded in recent years with the popularisation of accessible machine learning and artificial intelligence technologies[1]. These have applications in a great many areas, from diagnostic medicine to written language translation . Their varied uses demonstrate computer intelligence's utility in a number of problem domains, including computational creativity.

Creativity can be defined as

> The ability to transcend traditional ideas, rules, patterns, relationships, or the like, and to create meaningful new ideas, forms, methods [and] interpretations.

The traditional conception of computers as rigid, logic following machines could not be further from the above definition of creativity or notions of human artists who are renowned for the free flowing connection of ideas and concepts[2]. However the implementation of algorithms in non-traditional use cases and attempts to allow computers to learn independently have narrowed this gap significantly[3], facilitating the development of a spectrum of creative computing software. From more rigid, context aware systems like CAST to LSTM RNNs[4] like Benjamin which are discussed in Chapter 3. The world of narrative and text generation features many examples of this software and the variety of artefacts produced shows how impactful the level of constraints imposed during generation[5] are on the output - and subsequent creative merit - of the software and its works.

---

[1]See Tensorflow as an example of consumer friendly machine intelligence software .

[2]See and Chapter 3 for reference to historical creators famous for transforming conceptual spaces and discussion on what type of creativity is historically valued.

[3]See The Painting Fool and its continued advances .

[4]Long Short-Term Memory Recurrent Neural Networks.

[5]Where on the spectrum from rigid, goal oriented software, to less restricted, less context aware machine learning applications a system could be classified. See Section TKTKTK for more detail.