Creative Coding for Humor Design: A Preliminary Exploration

Alessandro Valitutti

Department of Computer Science and HIIT,
University of Helsinki, Finland

1 Introduction

Computational humor generation is a tricky problem. The idea to have a computer program capable of making people laugh, in a similar way today machines can beat them at chess, is intriguing. Unfortunately the prototypes built in the last decade do not seem to have exhibited humorous skills comparable to those of humans.

There are several factors contributing to the limitations in the current state of the art. First, good humor requires a large amount of common sense and linguistic knowledge. It is necessary for modeling (and then violating) people's expectations and inducing surprise, or for representing ridiculous or embarrassing events in an indirect and original way. Moreover, humor is a creative process. Most of jokes seem to be based on a linguistic invention or the discovery of a new funny association. For instance, the capability to quickly discover humorous associations is crucial in specific contexts, as in the case of the appropriateness of a witty remark during a conversation. Creativity provides humor with a dynamic dimension and a second order of complexity. Finally, the fruit of the creative discovery become part of the collective knowledge and can be reused as a cultural object.

These problems might be sufficient to discourage any significant advancement. Nevertheless there is another area of computer technology in which the above issues seem to be addressed in an unexpected way. It is named with different terms such as *generative art* or *creative coding*. Both humor and art aims are creative and complex processes, and both aim to induce an emotional effect (mirth in the former case, aesthetic pleasure in the latter one). In recent years a number of programming languages and computational environment for creative coding have been diffused and have attracted a growing community of artists and enthusiasts.

This contribution is a small collection of observations emerged during an explorative study of these environments, performed during the last months. It is aimed to search new ideas and methodological directions for the future development of research in computational humor generation.

2 The Knowledge Home

A few years ago the artificial life theorist Tommaso Toffoli proposed his vision of a *Knowledge Home*, a possible computational knowledge environment used as an extension of human mind (Toffoli, 2002). While resembling Dawkings' notion of "extended phenotype" (Dawkins, 1990), this metaphor emphasizes the role of knowledge sharing and reuse as a way to produce culture.

In less than one decade, a large part of this vision seems to have become real. Social networks, cloud computing, wikis, and smartphone apps are only a few examples of the multiplicity of innovations that are dramatically changing the nature of social relationships and allowing people to produce new forms of culture. In this transformation an important role is played by the new way to look at programming languages. They are not only associated to their syntax and semantics but also to the libraries developed and made available over the years. Java is a typical example of this change of perspective. Due to its high portability and suitability for Internet programming, it is continuously enriched by new packages, which are then included in the official distributions.

3 Creative Coding and Generative Art

A further step in the evolution of the computational cultural environments appears to be the recent development of environments for creative coding. This term is generally indicating the use of computer programming for building digital artworks, such as images, animations, and interactive multimodal objects. At present there are several projects of this nature, the most diffused of which is *Processing* (Reas and Fry, 2007). It is a scripting language and a development environment built upon Java and widely used by artists, students, and researchers[1].

A key feature of this language is the simplification of instructions for defining and visualizing graphical objects. The project website allows programmers to make their code available to other users. The artworks can be executed online as Java applets and are sorted and shown in the homepage according to the users' rates. In this way, the programs (called *sketches*) are easily executed and reused to build more complex artifacts.

4 Building Blocks for the Humorous Environment

The observational exploration of the environments for digital art and visual design are proposed as a form of inspiration to shape the future of computational humor generation. In particular, two elements are identified here as having a crucial role in the transformation of a computational environment in a cultural engine supported by active communities: *knowledge sharing* and *reuse*.

In a recent work, a first tentative step was taken in order to integrate different computational humor tools (Valitutti, 2012). A lexical resource consisting of a collection of ambiguous lexical item was collected and then used as source for the production of three

---

[1]Other interesting analogous projects are *Nodebox*, developed in Python, and *Cinder*, built in C++.

types of humorous puns: punning riddles, funny acronyms and variation of familiar expressions via lexical replacement. In a more general creative and humor-oriented environment, a community of designer would add new types of humorous strategies and integrate new computational linguistic resources.

It is worthwhile to observe that, in the case of creative coding environments, the contribution of human creativity is not necessarily a limitation for the development of artworks based on completely automatic forms of machine creativity. For instance, a good number of interactive animations are implemented exploiting Artificial Intelligence models such as swarm particle optimization or cellular automata. The humorous environment could be exploited as a test bed for the exploration of new wordplays or the discovery of new humorous semantic associations.

The next step in the development of a computational humor environment would be the design of a scripting language not only capable to exploit the available resources, but also appealing enough to motivate users to play and perform new forms of creative coding.

References

Dawkins, R. (1990). *The Extended Phenotype: The Long Reach of the Gene.* Oxford U. Press.

Reas, C. and Fry, B. (2007). *Processing: A Programming Handbook for Visual Designers and Artists for Visual Designers and Artists.* MIT Press.

Toffoli, T. (2002). Knowledge home: Personal knowledge structuring in a computer world. White paper, Boston University.

Valitutti, A. (2012). Ambiguous lexical resources for computational humor generation. In *Proc. of the 4th International Conference on Agents and Artificial Intelligence (ICAART2012)*, pages 532–535, Vilamoura, Algarve, Portugal.