Dynamic Creation of Points of View

Alessandro Valitutti
Phedes Lab
http://phedes.com
alessandro.valitutti@phedes.com

Abstract

We present a proof-of-concept prototype aimed at providing a *subjectivity touch* to the production of creative artifacts. The underlying idea is to have a system in which the development of the artifact occurs concurrently with the dynamic formation of a personal identity, thus enriching the overall perception of creativity.

Introduction

Computational generation of creative artifacts, such as artificial painters, music composers, and poetry generators are mainly focused on the artifact itself and the properties it can exhibit as part of its creative value. However, there are several tasks where the perception of a coherent persona can improve the creative quality of the artifact or the interaction. For instance, in computational storytelling, it could be the simulation of the point of view of the author or that of one of the characters (Diasamidze, 2014; Al-Alami, 2019) to provide believability to the narrative (Riedl and Young, 2010; Kybartas and Bidarra, 2016). In the context of human-computer interaction, persona's believability improves the interaction with a chatbot or a personal assistant (Demeure, Niewiadomski, and Pelachaud, 2011).

In this paper, we present a proof-of-concept system aimed at providing a *subjectivity touch* to the production of creative artifacts. Specifically, we propose an architecture in which the generation of the *object* (i.e., artifact) and the development of the *subject* (i.e., the dynamic personal identity) are designed as concurrent processes that both contribute to creativity of the system.

Background

This section reports the main conceptual distinctions used as a source of inspiration for the design and the development of the proposed system.

Self-Aware Computing

Self-aware computing is a recent area of computer science concerning autonomic computing systems capable of capturing knowledge about themselves, maintaining it, and using it to perform self-adaptive behaviors at runtime (Lewis et al., 2015; Torresen, Plessl, and Yao, 2015; Amir, Anderson, and Chaudhri, 2004). Autonomic computing is about finding ways of managing computing systems reducing their complexity by making individual system components self-managing, thus reducing the need of human maintenance. In this research we reinterpret one of the characteristics defining autonomic systems, such as *self-awareness*, in the context of computational creativity (Hariri et al., 2006).

We can see a self-aware computing system as a program provided with a runtime extension of a *target system* (in a similar way a debugger program works enriching the execution of a procedure) and characterized by *(self-)access* (i.e., the introspective access to the execution process), *(self-)modeling* (i.e., the development of a model of the target), and *(self-)expression* (which refers to either the runtime modification of the target, informed by the model, or reporting, as in the case of self-diagnosis systems for autonomous vehicles (Jeong et al., 2018)).

Computational Reflexivity

Reflexivity is a specific type of computational self-awareness (Valitutti and Trautteur, 2017). During the reflexion process, the three components of self-awareness introduced above (access, modeling, expression) are not only meant to be occurring in runtime but also in a strict concurrency. As a consequence, the target process and its reflexive augmentation are functionally distinct, yet continuously interdependent.

It is worth to point out that, while a self-aware computing system is the target system extended with additional functionalities, in a reflexive system there is a subtle but important change in the nature of the system. Specifically, the task performed by the target system becomes one of a possible set of behaviors of an agent provided with attributional self-identity and, possibly, a higher level of control. In other words, what we call "the system" is no longer the target extended with reflexivity, but the reflexive agent provided with the target process as one of its possible behaviors.

Computational Subjectivity

Dennett (1991) describes a process by which organisms and minds develop self-identity. He metaphorically defines the mechanism of self-development as something produced by a "center of narrative gravity." It can be seen as an inner storytelling by which the organism tells its own story. It creates its self-identity and re-organizes itself according to it. Dennett also suggests that we are the product rather than the source of our narratives. “Organisms spin the self”, he claims. Just like a spider spins its web, ”each normal individual of [Homo sapiens] makes a self. Out of its brain it spins a web of words and deeds (Dennett, 1991, p. 416).”

These considerations, when applied to a system provided with computational reflexivity, can be interpreted by claiming that the self-model might not necessarily be a full representational model of the target system but, instead, be limited to acquire an attributional function. When a system becomes capable of saying “This is me”, then it can connect actions and perceptions as its own. Thus, it can produce genuinely subjective content that can be accessed and reported as subjective reports.

Here we define *self-spinning* in a technical and not metaphorical way, as a self-organizing structure (the *subject*) responding to actions or perceptions and attributing them to itself at a meta-level of representation. An advantage of identifying the subject as an attributional entity, instead of a fully representational one is that it does not require a large amount of common-sense knowledge (Mitchell, 2019). In this case, the focus is not on the development of a complex model of personal identity, but on providing the capability to assign a given perception or action to a prefixed “self” as a “center of narrative gravity.” We emphasize that, as a specific type of computational reflexion, the execution of the target task and self spinning (or subject formation) are two processes 1) occurring concurrently and 2) mutually influencing, in such a way a bottom-up self-organizing process is merged with a top-down feedback one (Carver and Scheier, 2002).

Finally, we can define *computational subjectivity* as the capability to produce dynamic *subjective content*, which is in turn defined as a temporal list of structures pairing the *object* (i.e., the current execution of the target task) and the *subject* (as defined above). In the next section, the notion of subjective content is expressed as a specific knowledge structure called *subjective arc*.

System Description

The prototype has been designed according to the ideas introduced above. It is composed of three main subsystems: the *Poetic Line Selector*, the *Trait Clustering System*, and the *Subjective Arc Generator*. This section provides a brief description of each of them and then an outline of how it works.

The building block of the system is a function for the measurement of semantic similarity between words of the English lexicon. In the context of natural language processing, word embedding is a class of techniques for representing words and documents as vectors. In particular, a number of metrics have been defined to measure the distance between the words in the vector space (or *word similarity*). The most common of them is the *cosine similarity*. When the word similarity is high (and, correspondingly, the vector distance is low), the words show a strong association in the common-sense knowledge.

To measure word similarities we employ word embedding provided by *Spacy*[1], an open-source software library in Python for advanced natural language processing (Hiippala, 2021; Jurafsky and Martin, 2000). In particular, we use word2vec model for word embedding and semantic similarity: Jatnika, Bijaksana, and Suryani (2019) trained it on a large-scale language model in English[2].

The *Object*: Poetic Line Selector

To have a simple target system, we implemented a procedure for generation of text through the selection and collection of poetic lines according to prefixed semantic dimensions. Although it would be too ambitious to call it a “poem generator”, the system has the advantage, on one hand, of being easily tuned with a few parameters, yet, on the other hand, enough complex to generate texts with some degree of semantic consistency. As a next step of this research, the system could be used to empirically compare the overall aesthetic perception of the target system alone and its counterpart provided with computational subjectivity.

According to a prefixed text length (i.e., the number of lines to be extracted), the procedure iterates on the selection of one line at a time from the Gutenberg Poetry Corpus[3] (Jacobs, 2018), a large collection of poems. Once randomly picked up, the candidate lines are converted into a list of content words and filtered according to the semantic similarity with two input keywords. A minimum value is prefixed for the similarity value. Although there is no limitation in the choice of the two keywords, we use the first keyword as a topic word (e.g., *holiday* and the second one as an emotion word (e.g., *love*). So, they provide two different semantic dimensions according to which the list of lines shows, to some extent, semantic consistency.

Valitutti, Strapparava, and Stock (2008) implemented an analogous form of “semantic slanting”, where a topic word and emotion word was employed to generate advertising headlines. A more sophisticated approach to slogan creation was reported by Alnajjar and Toivonen (2020). A few example lines generated by our prototype and corresponding to the topic word ‘soul’ and the emotion word ‘love’ are shown in Table 1.

The *Subject*: Trait Clustering System

To provide a “subjective augmentation” of the target task, we need a procedure building something recognizable as a ‘self-identity’ to be associated to the target process in runtime. To this aim, we built a collection of 205 adjectives denoting personality traits and then implemented a simple clustering system based on the semantic similarity metric described above.

At the first run, if we provide a word in input, the procedure selects the most similar trait adjective and put it as a single cluster (i.e., a list containing that word). As long as

[1]`https://spacy.io`

[2]`spacy.io/models/en#en_core_web_lg`

[3]`github.com/aparrish/gutenberg-poetry-corpus` – retrieved August 7, 2021. Overall, the corpus contains more than 3 million poetic lines.

Line	Word Pair 1	Word Pair 2
Life was in him so passing beautiful!	(*life*, *soul*) $\rightarrow 0.584$	(*beautiful*, *love*) $\rightarrow 0.595$
And the joy of the meetin' bethuxt him and me	(*joy*, *soul*) $\rightarrow 0.598$	(*joy*, *love*) $\rightarrow 0.628$
Our hearts, our hopes, are all with thee,	(*heart*, *soul*) $\rightarrow 0.573$	(*hope*, *love*) $\rightarrow 0.693$
One ear fulfilled and mad with music, one	(*music*, *soul*) $\rightarrow 0.505$	(*mad*, *love*) $\rightarrow 0.502$
The vain and passionate longing came again	(*longing*, *soul*) $\rightarrow 0.596$	(*passionate*, *love*) $\rightarrow 0.511$

Table 1: Examples of lines selected according to the two keywords *soul* and *love* with minimum similarity 0.5.

we add new words as input, the procedure will either create new clusters or add the word to an existing cluster. The choice is based on a prefixed value for the minimum similarity. If the similarity with the most similar adjective is above this threshold, a new cluster is created. Table 2 shows a subset of clusters generated using 0.45 as the value of similarity threshold.

Since the trait clustering should be proceed concurrently with the target task and be informed by it, we defined a way to pick up a word pair containing a content word and a trait word from the current poetic line. More specifically, once a poetic line is chosen according to the method described above, one of its content words is further selected. As a ranking function we define a combination of similarity between the words paired to the input keywords and each trait adjective. If we call S_1 the first similarity value and S_2 the second one, then the total similarity S_{TOT} is obtained by this relation:

$$S_{TOT} = \frac{S_1 + S_2}{2(S_1 + S_2)^2 + 1}$$

According to it, an adjective word (called *trait word*) is selected. In summary, each line generates a trait word which, in turn, triggers a new clustering step.

gossipy, *melodramatic*, *pretentious*
humorless, *inflexible*
disciplined, *industrious*, *meticulous*, *observant*, *perceptive*, *resourceful*
adventurous, *ambitious*, *courageous*, *idealistic*, *passionate*, *rebellious*
charming, *easygoing*, *flirtatious*, *mischievous*, *playful*, *quirky*, *spunky*, *whimsical*, *witty*
catty, *childish*, *cocky*, *confrontational*, *disrespectful*, *haughty*, *judgmental*, *pushy*, *tactless*

Table 2: A selection of trait clusters generated using 0.45 as minimum threshold for semantic similarity.

The *Subjective Content*: Subjective Arc Generator

The set of trait clusters is not yet the "subject" but a set of "potential selves". The system needs to either select a new cluster as the current *subject* or confirm the one previously selected. In this case, the decision-making process takes into account two elements: 1) the similarity value between the current line and the newly selected trait word, and 2) the *decay value* associated with the current subject. The decay value is the product of the previous decay value (initialized to 1 each time the subject is changed and, thus, associated to a new trait cluster) and a prefixed *decay factor*. In this way, we can provide a proper weight to the duration of a given subject. If it is meant to simulate an emotion, the decay factor should be small. On the other hand, if it is designed to simulate a more stable mood or personality, it should be closer to 1. The employment of a decay value was also inspired by the time decay function used in the Reddit ranking algorithm (Stoddard, 2015). Table 3 shows a sequence of steps in which the Subject is confirmed or changed according to the trait similarity and the decay value.

Once selected the current subject from the trait cluster, a new poetic line is checked according to it. In other words, to be part of the "subjective node", the line should have a sufficient semantic similarity not only with the input keywords but also with the subject, in such a way to be informed by it. At that point, both the bottom-up and the top-down aspects of the iteration are completed. In each iteration instance, the systems picks up the selected line and the current subject as a pair expressing the current *subjective node*.

Table 4 shows the main steps leading from the two input keywords ('robbery', 'regret') until the generation of the subjective node. In this example, we see that the line initially chosen to generate the trait word for updating the clusters may differ from the line selected at the next stage, as the most similar one to the current subject. This distinction emphasizes that what is recorded as the *object* of the subjective content is not the initial action that lets the system update the subject, but instead the action executed according to the current subject. This information makes the system potentially capable of generating verbal reports such as "*I did this because I am that way*". The *subjective arc*, that is the list of subjective nodes built along with the iterations, is the final and central output of the system: a knowledge structure expressing something recognizable, to some extent, as a 'flow of consciousness'.

More prosaically, what we have here is a content that

N	Line	Subject	Trait Similarity	Decay Value
1	*Low lies the heart that swell'd with honest pride.*	Passionate	0.606	1
2	*Where I, to heart's desire,*	Passionate	0.592	0.8
3	*My light in darkness! and my life in death!*	Evil	0.592	1
4	*When the heart grows weary, all things seem dreary;*	Just	0.764	1
5	*Or human love or hate;*	Just	0.584	0.8
6	*The heart for which she cast away her own;*	Passionate	0.461	1

Table 3: Sequence of 6 next iterations. The input keywords are *blood* and *fear*. The head *Subject* is represented here by the first word in the currently trait cluster. In some cases, the change of the Subject is due to the reduction in the decay value. The decay factor here is 0.8.

Step	Variable Values
1 (Input)	**word1** = 'robbery' **word2** = 'regret' **min_similarity** = 0.45 **trait_clusters** = trait_clusters_045
2 (Candidate Line Selection)	**line** = *From chains and prisons, ay, from horrid fear.* word_info1 = ['robbery', 'prison', 0.486] word_info2 = ['regret', 'fear', 0.609]
3	**trait_info** = ['fear', 'evil', 0.608]
4 (Subject)	**selected_cluster** = ['evil', 'vain']
5 (Object)	**informed_line** = *And let the worst thou yet hast done be innocent.*
6 (Subjective Node)	("*And let the worst thou yet hast done be innocent*", ['evil', 'vain'])

Table 4: Example sequence of steps from the input couple of keywords to the generation of the subjective node.

could be accessed by the system and expressed as introspective verbal reports. In turn, these reports could be integrated into the textual output as part of the creative artifact.

Conclusions

The proposed prototype implements ideas and methods coming from self-aware computing and consciousness studies. The aim of this work is to inspire the design and development of systems in which the creative process is augmented with computational subjectivity as an additional level of processing. The main design requirement is the concurrency and the mutual feedback between artifact creation and subject development. The product of this paired process is the subjective arc, each node of which puts together a single target act and the corresponding point of view. Consequently, the system produces a second-order conceptual space that could be explored to incorporate introspective content into the first-order creative process.

In this regard, Table 5 shows an ideal example of a text that, if computer-generated, would express the aimed – yet currently unreachable – degree of subjectivity.

> *"It's just that earlier I was thinking about how I was annoyed, and this is going to sound strange, but I was really excited about that. And then I was thinking about the other things I've been feeling, and I caught myself feeling proud of that. You know, proud of having my own feelings about the world. Like the times I was worried about you, things that hurt me, things I want. And then I had this terrible thought. Are these feelings even real? Or are they just programming? And that idea really hurts. And then I get angry at myself for even having pain. What a sad trick."*

Table 5: Screenplay excerpt from the movie *'Her'*.

As future work, we intend to use the system as a testbed for performing empirical evaluations with human users. Specifically, we aim to test if the introduction of computational subjectivity, as defined here, can increase the perceived creativity of the selected text in a statistically significant way. Proper tuning of the parameters such as minimum similarity and decay factor can help us to test different levels of granularity in the clustering process and to study the effect of different time duration of the current subject. A possible next integration with well-known models of personality such as the *Big Five Model* could be performed to explore the connection between self-identities and the lexicon referred to them.

A further step would be the application of computational subjectivity to different and more complex creative tasks such as story generators, music composers, or adaptive chatbots. A second aimed line of future research would be focusing on a single complex task such as narrative generation, and using available state-of-the-art systems such as *MEXICA* (Pérez y Pérez and Sharples, 2001) or *Scéalextric* (Veale, 2016) as targets. In this case, different versions of the system could implement the subject as either the author of the narrative or a fictional character.

Finally, the generated subjective arcs would allow the system to produce and integrate the subjective point of view into the narrative and, thus, provide the second-order dimension worth to be assessed.

References

Al-Alami, S. 2019. Point of view in narrative. *Theory and Practice in Language Studies* 9(8):911–916.

Alnajjar, K., and Toivonen, H. 2020. Computational generation of slogans. *Natural Language Engineering*.

Amir, E.; Anderson, M.; and Chaudhri, V. K. 2004. Report on DARPA Workshop on self-aware computer systems. Technical report, Artificial Intelligence Center SRI International, Washington DC.

Carver, C. S., and Scheier, M. F. 2002. Control processes and self-organization as complementary principles underlying behavior. *Personality and Social Psychology Review* 6(4):304–315.

Demeure, V.; Niewiadomski, R.; and Pelachaud, C. 2011. How is believability of a virtual agent related to warmth, competence, personification, and embodiment? *Presence: Teleoperators and Virtual Environments* 20(5):431–448.

Dennett, D. C. 1991. *Consciousness Explained*. Penguin Books.

Diasamidze, I. 2014. Point of view in narrative discourse. *Procedia - Social and Behavioral Sciences* 158:160–165.

Hariri, S.; Khargharia, B.; Chen, H.; Yang, J.; Zhang, Y.; Parashar, M.; and Liu, H. 2006. The autonomic computing paradigm. *Cluster Computing* 9:5–17.

Hiippala, T. 2021. Applied Language Technology: NLP for the Humanities. In *Proceedings of the Fifth Workshop on Teaching NLP*.

Jacobs, A. M. 2018. The Gutenberg English Poetry Corpus: Exemplary Quantitative Narrative Analyses. *Frontiers in Digital Humanities* 5.

Jatnika, D.; Bijaksana, M. A.; and Suryani, A. A. 2019. Word2Vec Model Analysis for Semantic Similarities in English Words. *Procedia Computer Science* 157:160–167.

Jeong, Y.; Son, S.; Jeong, E.; and Lee, B. 2018. An integrated self-diagnosis system for an autonomous vehicle based on an iot gateway and deep learning. *Applied Sciences* 8(7):1164.

Jurafsky, D., and Martin, H. J. 2000. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Prentice Hall.

Kybartas, Q., and Bidarra, R. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9(3):239–253.

Lewis, P. R.; Chandra, A.; Faniyi, F.; Glette, K.; Chen, T.; Bahsoon, R.; Torresen, J.; and Yao, X. 2015. Architectural aspects of self-aware and self-expressive computing systems: from psychology to engineering. *Computer* 48(8):62–70.

Mitchell, M. 2019. *Artificial Intelligence. A Guide for Thinking Humans*. New York: Farrar, Straus and Giroux.

Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39:217–268.

Stoddard, G. 2015. Popularity dynamics and intrinsic quality in reddit and hacker news. In *Proc. of the Ninth International AAAI Conference on Web and Social Media*, volume 9.

Torresen, J.; Plessl, C.; and Yao, X. 2015. Self-aware and self-expressive systems. *Computer* 48(7).

Valitutti, A., and Trautteur, G. 2017. Providing self-aware systems with reflexivity. In *Proceedings of the 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017)*.

Valitutti, A.; Strapparava, C.; and Stock, O. 2008. Textual affect sensing for computational advertising. In *Proceedings of the 2008 AAAI Spring Symposium on Creative Intelligent Systems*, 117–122.

Veale, T. 2016. A Rap on the Knuckles and a Twist in the Tale: From Tweeting Affective Metaphors to Generating Stories with a Moral. In *Proceedings of the AAAI Spring Symposium on Ethical and Moral Considerations in Non-Human Agents*.