Nardine Osman, Matthew Yee-King (Eds.)

AI and Feedback

First International Workshop, AInF 2015
Buenos Aires, Argentina
July 25–27, 2015

CEUR Workshop Proceedings

Volume Editors

Nardine Osman
IIIA – Artificial Intelligence Research Institute
CSIC – Spanish Scientific Research Council
Campus de la Universitat Autonoma de Barcelona
08193 Bellaterra, Spain
E-mail: nardine@iiia.csic.es

Matthew Yee-King
Department of Computing
Goldsmiths, University of London
London SE14 6NW
United Kingdom
E-mail: m.yee-king@gold.ac.uk

Published by the editors on CEUR-WS.org

CEUR-WS.org ISSN 1613-0073 Volume 1407 http://CEUR-WS.org/Vol-1407

July 2015

Preface

This volume of the CEUR Workshop Proceedings contains papers accepted for the First International Workshop on AI and Feedback (AInF 2015), held in Buenos Aires, Argentina, July 25–27, 2015. This workshop was co-located with the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015).

Feedback is key for both improvement and decision making. As humans, we are designed to constantly seek feedback on how and what we are doing in life. Feedback can come from ourselves, from our peers, from our teachers, from our collaborators, audiences, customers, public or press. Feedback provides opportunities to learn about how we and our work are perceived by others. If we encounter someone [something] new, we can examine previous feedback to learn how this new person [thing] is perceived by others.

The aim of the AI and Feedback workshop is to motivate research that focuses on applying AI techniques for addressing the challenges of mining and extracting feedback, as well as assessing, analysing, and making use of feedback. Furthermore, a key target of the AI and feedback field in general should be to investigate how to build intelligent feedback agents that are capable of autonomously providing feedback that equals or surpasses that of human beings in its usefulness. The feedback of artificial feedback agents should have some desirable characteristics. It should be socially and culturally appropriate, clearly expressed, sufficiently focused and contextualised, thoughtfully challenging yet encouraging, compassionate, open to debate, justified and comparative, also, it should be trustworthy. Giving and receiving feedback with these characteristics therefore is a challenging, creative process.

This is the very first international workshop on the topic. We hope it will provide the motivation needed to advance research on this interesting and influential topic.

July 2015

Nardine Osman
Matthew Yee-King

Organisation

Workshop Chairs

Nardine Osman	Artificial Intelligence Research Institute, Spain
Matthew Yee-King	Goldsmiths, University of London, UK

Program Committee

James Bailey	University of Melbourne, Australia
Oliver Bown	University of Sydney, Australia
Mark d'Inverno	Goldsmiths, University of London, UK
Alice Eldridge	University of Sussex, UK
Jesualdo Tomás Fernández-Breis	University of Murcia, Spain
Ilya Goldin	Pearson Education, USA
Sumit Gulwani	Microsoft Research, USA
Sergio Gutierrez-Santos	University of London, UK
Sharon Hsiao	Arizona State University, USA
Jimmy Huang	York University, Canada
Anna Jordanous	University of Kent, UK
Chris Kiefer	University of Sussex, UK
Manolis Mavrikis	University College London, UK
François Pachet	Sony Computer Science Laboratory, France
Emanuele Ruffaldi	Perceptual Robotics Laboratory, Italy
Dario Sanfilippo	Freelance
Carles Sierra	Artificial Intelligence Research Institute, Spain
Luc Steels	Vrije Universiteit Brussel, Belgium
Shusaku Tsumoto	Shimane University, Japan
Jacco van Ossenbruggen	Centrum Wiskunde and Informatica, The Netherlands
Toby Walsh	NICTA & University of New South Wales, Australia
Daniel Zeng	University of Arizona, USA

Contents

Praise or flow? Two pedagogies for open-ended learning

— Invited Speaker's Abstract —

Luc Steels
ICREA (IBE, UPF/CSIC), Barcelona, Spain

Abstract

MOOCs have recently flourished as a new way to bring education to large numbers of people at affordable cost. However most MOOCs so far rely on rigid structured instruction based on prior lesson plans. Can we also develop MOOCs that follow the paradigm of open-ended, student-centered learning? This requires (i) a challenging environment and tools in which students can learn how to solve problems without a rigid prior lesson plan, (ii) ways in which to orchestrate peer-to-peer social feedback between students, and (iii) mechanisms fostering motivation. This talk focuses on the latter. I discuss two pedagogies at opposite ends of a spectrum: one based on praise, which means encouragement or possibly punishment, the other based on flow, which means that students can regulate their own problem challenge in relation to their skill level and thus become self-motivated.

About the invited speaker

Luc Steels studied linguistics at the University of Antwerp (Belgium) and computer science at the Massachusetts Institute of Technology (USA). His main research field is Artificial Intelligence covering a wide range of intelligent abilities, including vision, robotic behavior, conceptual representations and language. In 1983 he became a professor of computer science at the University of Brussels (VUB) and in 1996 he founded Sony Computer Science Laboratory in Paris and became its first director. Currently he is an ICREA Research Professor at the Institute for Evolutionary Biology (CSIC, UPF). He has been the PI of a dozen large-scale European projects and more than 30 PhD theses have been granted under his direction. He has produced over 200 articles and edited 15 books related to his research.

Pedagogical agent models for massive online education

Matthew Yee-King and Mark d'Inverno
Department of Computing, Goldsmiths College
London, UK
m.yee-king@gold.ac.uk

Abstract

The effective implementation of Massively Open Online Courses poses fascinating challenges. We address two such challenges using an agent based approach, employing formal specifications to articulate an agent design which can later be used for software development. The challenges addressed are: 1) How can a learner be provided with a personalised learning experience? 2) How can a learner make best use of the heterogenous community of humans and agents who co-habit the virtual learning environment? We present formal specifications for an open learner model, a learning environment, learning plans and a personal learning agent. The open learner model represents the learner as having current and desired skills and knowledge and past and present learning plans. The learning environment is an online platform affording learning tasks which can be carried out by individuals or communities of users and agents. Tasks are connected together into learning plans, with pre and post conditions. We demonstrate how the personal learning agent can find learning plans and propose social connections for its user within a system which affords a dynamic set of learning plans and a range of human/ agent social relationships, such as learner-teacher, learner-learner and producer-commentator.

1 Introduction

2012 has been referred to as the 'year of the MOOC', the massive, open, online course [Pappano, 2012]. Indeed one of the authors of this paper was part of a team which delivered a course to an enrolled student body of around 160,000 in 2013 and 2014. The obvious problem with MOOCs is that there is a very high student to tutor ratio. This means it is not feasible to provide students with direct tutor support when they have problems with their learning and complex assessments which cannot be automated become impractical. The current solutions seem to be the use of forums and other social media wherein peer support can take place, and the use of peer assessment techniques such as calibrated peer assessment [Koller and Ng,]. Running our MOOC, we noticed that the forum seemed to be an inefficient tool through which students could find information, where the same questions would be asked and answered repeatedly, and where the constant churn pushed old answers away.[1] It was not clear if anyone would bother to answer a given question, or who would be the ideal person to answer it. Regarding the assessment, there was a tendency to assess others' work superficially - to simply fulfil the most basic requirements of the peer assessment task. This was probably an instance of strategic learning, where the learner does the minimum to meet the apparent requirements. Another problem is a high drop out rate on courses. For example, we had around 10% of our 150.000 students still active at the end of our MOOCs; Norvig and Thrun's famous Stanford AI CS211 course in 2011 went from 160,000 enrolments to 20,000 completions [Rodriguez, 2012]. These figures improve if we instead consider the number of students actively accessing learning materials at the start of the course; in our case, 100.000 becomes 36,000. So motivation to complete the course is another area that needs work.

But how might one motivate a learner, given the particular characteristics of a MOOC, i.e. the high learner to teacher ratio, the presence of a large, heterogeneous peer group, the distance, as opposed to on-campus learning aspect and so on? Might motivation be amplified by leveraging the learner's peers - the social network? What might a 'networked learner' gain from being part of an active learning community? How can the learner be made aware of the structure and members of the community, and how that might help them achieve their learning goals?

In summary, guidance for learners, feedback to learners (on their work) and general learner motivation are areas for improvement for MOOCs. These are the key points we aim to address in our wider research work. In this paper we present our work on a representative component of this: the invention of a type of pedagogical agent called a *personal learning agent* which can provide a more intuitive and efficient route through the learning materials and information, and which can help the learner to explore the network of other learners to find help or to provide help and feedback to others.

[1] this is somewhat alleviated by up-and down-voting of questions and answers but this is far from perfect

Pedagogical agents There is a significant literature around pedagogical agents and there are many questions one might ask when considering the design of pedagogical agents.

What is the purpose of the agent and is it pro-active, reactive, conversational or argumentative? Skiar et al. present a review of work where agents are used to supporting learning [Sklar and Richards, 2006]. The researchers define three main trends in the field: pedagogical agents, peer learning agents and demonstrating agents. According to Soliman and Guetl, Intelligent Pedagogical Agents (IPAs) can help learners by 'providing narrations ... creating adaptive dialogues with the learner to improve learning situations, provide guidance, resolve difficulties, and improve motivation' [Soliman and Guetl, 2010]. Quirino et al. implemented a case based reasoning driven IPA for training medical students. They define the following important characteristics: domain-specific knowledge, autonomy, communicability, learning, reactivity and pro-activity, social skills, customisation, and learning abilities [Quirino *et al.*, 2009]. Magus et al. describe a math tutoring game which includes a conversational agent [Magnus *et al.*, 2010]. They have explored aspects of the visual embodiment of the agent as well as its conversational capabilities. The conversation can occur in a a focused, on topic mode mediated through multiple choice questions and a free, off topic mode. Agents capable of argumentation have appeared in the education technology literature. In 2009, Tao et al. presented a pilot study where agents and learners engaged in learning through argumentation around the topic of food chains (e.g. tiger eats sheep eats grass) [Tao *et al.*, 2009]. The user interacts with the agent through keyboard, mouse and text to speech conversion (agent talks to learner) and the agent is capable of engaging in various types of dialogue. The researchers found preliminary evidence that the learners enjoyed interacting with the arguing agent.

Is the agent an animated character? Lester et al. trialled a 3D animated character with 100 middle school children. They discuss the *persona effect*, which encompasses the agent's encouragement (of learners), utility, credibility, and clarity, and which is much enhanced by the use of an animated character [Lester *et al.*, 1997]. In a subsequent survey of animated pedagogical agents, Johnson et al. provide a list of technical issues for designers of animated pedagogical agents to consider: interface to the environment, behavioural building blocks, behaviour control. believability, emotion, platform and networking Issues [Johnson *et al.*, 2000].

How competent is the agent and what is its role? Xiao et al. empirically assessed the effect of pedagogical agent competency where learners were learning how to use a text editor supported by pedagogical agents with varying competency at the task [Xiao *et al.*, 2004]. Allowing users to choose the competency of their agent improved objective performance. Kim and Baylor present a study investigating the value of pedagogical agents as learning companions (PALs) with deliberately varying competency levels and interaction modes. They conclude that 'PALs should be designed as highly competent for learning contexts in which instructional goals focus on knowledge and skill acquisition...in contexts where learners' self-efficacy beliefs in the task are a major concern, less competent PALs could be more effective'[Kim and Baylor, 2006]. Baylor et al. present an initial study where agents take on different roles when supporting learners: Motivator, Expert, or Mentor. More knowledgeable agents were more credible and seemed to transfer more knowledge but motivating agents were more engaging [Baylor and Group, 2003].

What are the requirements for a pedagogical agent in a large scale, social learning context? Leading towards our interest in social learning, in [Spoelstra and Sklar, 2007], an agent based approach is used to simulate interactions between learners within a group. A parameterised learner model is presented which includes features such as ability, emotion, motivation (inc. competitiveness), learning rate, understanding, 'likeliness to help' and progress. Instances of the model are run in simulation and characteristics observed in real groups of learners are observed, such as the importance of group composition, team size and team rewards.

Research questions We have discussed our wider research goals in the introduction: better pathways to and through information for learners, better feedback to learners (on their work) and increasing learner motivation. In this paper, we will address the following questions which fall within this wider remit: How might one formally specify a human learner to allow operations upon that information by an autonomous agent? What kind of operations might be useful, given the wider research goals?

2 Agent requirements

We will begin by framing the agent specification presented later with some requirements for the functionality of the agent. There are 4 key requirements: to store learner state, to report learner state, to find learning plans and to propose social connections. Each of these requirements has sub-requirements, as listed below:

1. Storing learner state:
   (a) Storing the goals of a person
   (b) Interpret the goals into desired skills and knowledge
   (c) Storing a person's current skills and knowledge
   (d) Storing a person's current and previous plans
2. Reporting learner state:
   (a) Reporting current state of goals and plans
   (b) Reporting current state of knowledge and skills
   (c) Reporting status of data/ content provided to and from the community i.e. plans, feedback, feedback agents, trust model
3. Plan finding:
   (a) Propose plans whose pre-conditions match current skills and knowledge
   (b) Propose plans whose post-conditions (goals) match a persons goals
   (c) Generate evaluation data for plans based on users
   (d) Propose plans which are successful, i.e. verified post conditions

4. Agent finding:
   (a) Propose social relationships/ connections to people with similar goals/ skills/ knowledge (potential peers, potential as they must actively agree to connect to make a social relationship)
   (b) Propose connections to people with similar (musical/ geographical/ etc.) data
   (c) Propose connections to people who have related but superior skills and knowledge (potential tutors), or teaching goals. (I want to increase others' knowledge of scales on the guitar). These people might be able to assign plans, for example.

3 Formal specification

In this section we will use the specification language Z to develop the models of our agents, following the methodology developed by Luck and d'Inverno [D'Inverno and Luck, 2003; Luck and D'Inverno, 1995].

Learner model

We begin our description by introducing our learner model. The purpose of the learner model is to represent various aspects of a person operating within our learning environment. There are two *types* which users of the system might want to learn about or teach about. The specification remains neutral about how they are encoded but this encoding might include free text descriptions or formulae in predicate calculous, for example.

$$[Skill, Knowledge]$$

As an example, a user might have the skill of playing the C major scale and the knowledge which includes being able to state which notes are in the scale of C major.

We then define $Proficiency$ as the combination of skills and knowledge, representing all that a person would potentially wish to learn in music.

$$Proficiency ::= skills\langle\langle Skill\rangle\rangle \mid knowledge\langle\langle Knowledge\rangle\rangle$$

A particular person can be given a score which is an evaluation of their learning level regarding a particular skill or knowledge element:

$$Score == \mathbb{N}$$

Learning environment

We continue the description with some details about the learning environment which learners, teachers and agents will inhabit. For the purposes of our wider research, it is specialised for music education, and it is designed around a social, blended learning pedagogy wherein people upload recordings of themselves playing instruments and other media items. Discussion and feedback can occur around the uploaded items. Within the environment, people and agents can carry out tasks, where a task is something to be undertaken.

$$[Task]$$

We have identified 9 distinct tasks which can be carried out within our learning environment.

$$TaskType ::= Practice \mid Listen \mid Makemusic \mid Upload \mid Share \mid Annotate \mid Question \mid Answer \mid Visualise$$

Earlier, we mentioned that feedback might be provided about a media item. For the time being we define feedback as a given set. It is possible to define feedback in terms of constructive and evaluative praise and criticism. However, these are our first attempts at defining feedback and we will remain neutral for the time being.

$$[Feedback]$$

We define evaluate to be a function which maps an proficiency to a natural number, e.g. 'I have evaluated the way you have played C major as scoring a 5'.

$$\mid evaluateproficiency : Proficiency \to \mathbb{N}$$

In the system the community may evaluate many different aspects, such as feedback for example.

$$\mid evaluatefeedback : Feedback \to \mathbb{N}$$

Goals, Beliefs and Plans

As with the definition of the SMART Agent Framework [D'Inverno and Luck, 2003] we take a goal to be a state of affairs in the world that is to be achieved (by some agent).

$$[Goal]$$

The way that goals (or, equally, learning outcomes) are achieved is through a workflow of tasks: a sequence of tasks that have to be completed in order. We do not specify here who determines whether the tasks have been accomplished successfully or not because in general this could be a mixture of the system, the user themselves, the community and/or a teacher. Plans are typically specified in terms of what must be true before they can be adopted, what is true after they have been successfully completed, and the kinds of actions (or in our language tasks) that have to be completed in order. Next we define a plan to be a set of preconditions (the skills and knowledge an agent must have before undertaking the plan) and a set of post conditions (the new set of skills and knowledge the agent will have after the plan). The predicate part of the schema states that the intersection of the pre and post conditions is necessarily empty.

Plan

$$pre : \mathbb{P}\ Proficiency$$
$$post : \mathbb{P}\ Proficiency$$
$$workflow : \text{seq}\ Task$$
$$pre \cap post = \{\}$$

In specifying this system, it is useful to be able to assert that an element is optional. The following definitions provide for a new type, $optional[T]$, for any existing type, T, which consists of the empty set and singleton sets containing elements of T. The predicates, $defined$ and $undefined$ test whether an element of $optional[T]$ is defined (i.e. contains an element of type T) or not (i.e. is the empty set), and the

function, *the*, extracts the element from a defined member of $optional[T]$.

$$optional[X] == \{xs : \mathbb{P}\, X \mid \# xs \leq 1\}$$

$$[X]$$
$$defined\_, undefined\_ : \mathbb{P}(optional[X])$$
$$the : optional[X] \pfun X$$
$$\forall xs : optional[X] \bullet$$
$$\quad defined\, xs \Leftrightarrow \# xs = 1 \land$$
$$\quad undefined\, xs \Leftrightarrow \# xs = 0$$
$$\forall xs : optional[X] \mid defined\, xs \bullet$$
$$\quad the\, xs = (\mu\, x : X \mid x \in xs)$$

$$Bool ::= True \mid False$$

Using this definition we can now specify the state of a plan. The state of a plan can be thought of as a running instance of a plan during the lifetime of a user's activity. It means that the plan has been adopted to achieve a goal. In order to specify this we keep the information contained in the specification of a plan using schema inclusion. We also state that if the plan has been started but not finished there will be a *current task* that the agent is currently undergoing. The predicate part states that the current task must have been defined in the workflow of the plan. By also defining a flag called *finished* we can specify a plan state as follows.

PlanInstance
$$Plan$$
$$current : optional[Task]$$
$$finished : Bool$$
$$thecurrent \in (\mathrm{ran}\, workflow)$$

The initial plan state (for any state schema the initial state should be specified in Z) is where the plan has just been proposed or adopted by a user.

InitialPlanInstance
$$PlanInstance$$
$$undefined\, current$$
$$finished = False$$

We are now in a position to define four specific sub-types of the plan state as follows.

1. Proposed Plan. A plan which has been selected to achieve a goal but which has not been started by the agent. As no task has been started the current task is set to undefined.

   ProposedPlan
   $$InitialPlanInstance$$

2. Active Plan. A plan which is ongoing. It has not been completed and the current task is set to defined.

   ActivePlan
   $$PlanInstance$$
   $$defined\, current$$
   $$finished = False$$

3. FailedPlan. This is a plan which has a defined task but a flag set to finished. For example, this represents a situation where one of the tasks in the workflow of a plan is too difficult for the user and the plan is abandoned by the user.

   FailedPlan
   $$PlanInstance$$
   $$definedcurrent$$
   $$finished = True$$

4. Completed Plan. The flag *finished* is set to true and the current task becomes undefined.

   CompletedPlan
   $$PlanInstance$$
   $$undefined\, current$$
   $$finished = True$$

There are several operations that we could specify at the level of the plan but the key one is finish task. Either this leads to the plan being completed or the current place in the work flow moves to the next task.

In the first case the specification looks like this:

FinishTask1
$$\Delta PlanInstance$$
$$current = \{last(workflow)\}$$
$$finished = False$$
$$undefined\, current'$$
$$finished' = False$$

In the second case like this:

FinishTask2
$$\Delta PlanInstance$$
$$current \neq \{last(workflow)\}$$
$$finished = False$$
$$current' = \{workflow((workflow^{\sim}(the\, current)) + 1)\}$$
$$finished' = False$$

The other is to instantiate a plan which essentially means creating a PlanInstance in its initial state from a Plan.

$$instantiateplan : Plan \to InitialPlanInstance$$
$$\forall p : Plan;\ ps : InitialPlanInstance$$
$$\quad \mid ps = instantiateplan(p) \bullet$$
$$\quad ps.pre = p.pre \land ps.post$$
$$\quad = p.post \land ps.workflow = p.workflow$$

The (almost) inverse function of this is a function which takes any PlanInstance and returns the plan.

$$recoverplan : PlanInstance \to Plan$$
$$\forall p : Plan;\ ps : PlanInstance \mid p = recoverplan(ps) \bullet$$
$$\quad ps.pre = p.pre \land$$
$$\quad ps.post = p.post \land ps.workflow = p.workflow$$

Beliefs

This is a representation of what the agent knows and what it can do. Again we remain neutral on the representation.

$[Belief]$

The Personal Learning Agent

In the schema below we have the following definitions:

1. An agent has a set of goals at any stage which we call desires (typically these are associated with learning outcomes as described earlier in the document).
2. An agent has a set of beliefs. These refer to the information which is stored about what the user knows or what the user can do (skills).
3. An agent has an interpret function which takes a goal and returns a set of proficiencies (skills and knowledge). Note that the complexity of this function may vary as in some cases goals may be expressed as a set of proficiencies directly and so this function becomes a simple identity function. However, in other situations this function has to take a free text description and turn it into a set of proficiencies. Clearly, in general no automatic process can do this and such an operation will often be left to the community. In which case we specify the agent's interpret function as a *partial* function.
4. An agent has a similar interpret function for beliefs which maps its beliefs to a set of machine readable (skills and knowledge).
5. $intdesires$ is a set of proficiencies which can then be used by the agent and the community to plan. Note then, that $interpreteddesires$ is made up of the automatic function $interpret$ of the agent, possibly the automatic interpretation of other agents, but also from human users in the music learning community.
6. $intbeliefs$ is the analagous set of proficiencies which the agent has recorded as known or accomplished by the agent.
7. It is not unreasonable to suggest that all tasks are not available to a user at all times and so the agent can record which tasks are currently available to a user. (If a user is offline, upload is not an available task. If a newcomer joins a community then possibly they do not feel like giving any feedback and so the agent can record that the user is currently not offering this task.).
8. Then we define the set of plans which the agent knows about (possibility learned from other agents). This is where the agent contains its *procedural knowledge* about what plans work in what situations to achieve which desired proficiency.
9. The agent maintains a record of all of the plans that have been completed and all of those which have failed.
10. There is a record of the intentions. This is a mapping from a set of proficiencies (this set may only have one proficiency in it of course) to the plan instance which the agent has adopted to attain those proficiencies.
11. Finally, we record all those interpreted desires for which the agent has no active plan.

    There are also two dummy variables that we can use (which can be calculated from the variables described so far but which aid us in the readability of the specification)

12. We define a variable to store the tasks that the agent is currently involved in ($currenttasks$) which can be calculated as the union of the tasks from the current plans.
13. We define a variable to store the current plan instances of the agent

Next we consider the constraints on the state of a personal learning agent

1. The interpreted desires are the result of applying the interpret desire function to the desires.
2. The interpreted beliefs are the result of applying the interpret desire function to the beliefs.
3. The intersection between interpreted desires and interpreted beliefs is an empty set, (in other words you can't desire a proficiency you already have).
4. If there is a plan for a subset of proficiencies then those proficiencies must be contained in the the interpreted desires.
5. If there is a plan for one subset of proficiencies and a plan for another distinct set of proficiencies then their intersection is empty.
6. The unplanned desires are those interpreted desires for which there is no intention.
7. The current tasks are calculated from iterating the current plans and accumulating the current tasks for each plan.
8. The current plans are calculated by taking the range of the intentions.

$$
\begin{array}{|l}
\hline
[X, Y] \\
\hline
map : (X \to Y) \to (\operatorname{seq} X) \to (\operatorname{seq} Y) \\
mapset : (X \to Y) \to (\mathbb{P}\, X) \to (\mathbb{P}\, Y) \\
\hline
\forall f : X \to Y;\ x : X;\ xs, ys : \operatorname{seq} X \bullet \\
\quad map\, f\, \langle\rangle = \langle\rangle \land \\
\quad map\, f\, \langle x \rangle = \langle f\, x \rangle \land \\
\quad map\, f\, (xs \frown ys) = map\, f\, xs \frown map\, f\, ys \\
\forall f : X \to Y;\ xs : \mathbb{P}\, X \bullet \\
\quad mapset\, f\, xs = \{x : xs \bullet f\, x\} \\
\hline
\end{array}
$$

PersonalLearningAgent

$desires : \mathbb{P}\ Goal$
$beliefs : \mathbb{P}\ Belief$
$interpretdes : Goal \pfun \mathbb{P}\ Proficiency$
$interpretbel : Belief \pfun \mathbb{P}\ Proficiency$
$intdesires : \mathbb{P}\ Proficiency$
$intbeliefs : \mathbb{P}\ Proficiency$
$availabletasks : \mathbb{P}\ TaskType$
$plandatabase : \mathbb{P}\ Plan$
$completedplans, failedplans : \mathbb{P}\ Plan$
$intentions : (\mathbb{P}\ Proficiency) \pfun PlanInstance$
$unplannedintdesires : \mathbb{P}\ Proficiency$

$currenttasks : \mathbb{P}\ Task$
$currentplaninstances : \mathbb{P}\ PlanInstance$

$intdesires = \bigcup(mapset\ interpretdes\ desires)$
$intbeliefs = \bigcup(mapset\ interpretbel\ beliefs)$
$intdesires \cap intbeliefs = \emptyset$
$\bigcup(\mathrm{dom}\ intentions) \subseteq intdesires$
$\forall ps1, ps2 : \mathbb{P}\ Proficiency \mid (ps1 \neq ps2) \land (\{ps1, ps2\} \subseteq (\mathrm{dom}\ intentions)) \bullet ps1 \cap ps2 = \{\}$
$unplannedintdesires = \bigcup(\mathrm{dom}\ intentions) \setminus intdesires$

$currenttasks = \{t : Task;\ ps : PlanInstance \mid ps \in (\mathrm{ran}\ intentions) \bullet the\ ps.current\}$
$currentplaninstances = \mathrm{ran}\ intentions$

Plan Finding

Plan finding is the process of taking a set of candidate plans and selecting those whose preconditions are met and where at least some subset of the postconditions are desired.

For this operation we assume the input of a set of candidate plans. Again we do not specify whether these come from the agent (i.e. the agent's database of plans), other agents in the community, from the user or from other users. In general, candidate plans with be a *synthesis* of the users and the agents of users working together.

For now we will suppose that suitable plans have all pre-conditions satisfied and it is the case that both: (a) none of the postconditions are things which the user is already proficient in (b) all of the postconditions are current interpreted desires of the user. In the schema below *SuitablePlans* is generated which satisfies this constraint and from these one plan *adoptedplan* is selected. The state of the agent is then updated such that its current plans include a mapping from the pre-conditions of the plan (which are necessarily interpreted desires for which no plan exists).

FindandAdoptPlan

$PossiblePlans? : \mathbb{P}\ Plan$
$SuitablePlans! : \mathbb{P}\ Plan$
$adoptedplan : Plan$
$\Delta PersonalLearningAgent$

$SuitablePlans! = \{ps : PossiblePlans? \mid (ps.pre \subseteq intbeliefs) \land (ps.post \cap unplannedintdesires) = \{\} \bullet ps\}$
$adoptedplan \in SuitablePlans!$
$intentions' = intentions \cup \{(adoptedplan.post, instantiateplan(adoptedplan))\}$

It would be a simple matter to add more detail to this schema including choosing the plan with the highest rating for example, or a plan which has completed successfully in the community the most number of times, or making sure the plan has not failed in the users history, or that the plan has not failed in the community with users which have similar profiles as defined by the personal learning agent. In general, the plan finding system requirements, and this specification alongside it, will develop as we gain experience of how the system is used.

Plan Completion

The very simplest way this could happen is as follows:

1. Because of a successfully completed task a plan instance becomes an element of $CompletedPlan$.
2. The post conditions are added to the interpreted beliefs (these may in turn be reverse ineterprered into beliefs which can then be seen by the community).
3. Any post conditions that were formerly desires are now removed from interpreted desires (these may in turn be reverse interprered into beliefs which can then be seen by the community).
4. The completed plans function is updated with the plan that has just successfully completed.

CompletePlan

$completedplan? : CompletedPlan$
$\Delta PersonalLearningAgent$

$completedplan? \in (\mathrm{ran}\ intentions)$
$intentions' = intentions \rhd \{completedplan?\}$
$intdesires' = intdesires \setminus completedplan?.post$
$intbeliefs' = intbeliefs \cup completedplan?.post$
$completedplans' = completedplans \cup \{recoverplan\ completedplan?\}$

However, this process will not be automatic in general within the system. In general, the user (or other users in the community) will be asked to evaluate the plan. There may be several ways in which this can happen. For example, a simple score could be given but in general each user who is evaluating the plan considers each of the post conditions (or another member of the community does) to work out whether they are now proficiencies (interpreted beliefs), whether they have not

been met and so are still interpreted desires, or whether they have not been met but are not desires. Indeed the evaluating user could rank each of the postconditions with a score and the agent may also wish to keep a snap shot of the agent's state for future comparison by the community.

Community of Music Learning

Agent finding

Now we move to defining a community of learners each of which has one and only one personal learning agent.

First we define the set of all users.

$$[User]$$

$$\begin{array}{|l} \textit{Community} \\ \hline community : \mathbb{P}\, User \\ agents : User \pfun PersonalLearningAgent \\ \hline community = \mathrm{dom}\, agents \end{array}$$

To this we can define all kinds of social relationships. For example, peer and teacher and others as they become useful. It is up to the designer of the system to state what the constraints are on any such relationships. To provide examples (not necessarily ones we would subscribe to) of how this is done we state that if user1 is a peer of user2 then user2 is a peer of user1 and, in addition, if user2 is a teacher of user1 then user1 cannot be a teacher of user2. Another example would be the idea of a fan who would always adopt the advice of another.

$$\begin{array}{|l} \textit{SocialRelationships} \\ \hline peer, teacher : User \rel User \\ fans : User \rel User \\ \hline \forall\, u1, u2 : User \bullet (u1, u2) \in \\ peer \Rightarrow (u2, u1) \in peer \\ \forall\, u1, u2 : User \bullet (u1, u2) \in \\ teacher \Rightarrow (u2, u1) \notin teacher \end{array}$$

Using these schemas it then becomes possible to ask agents to start to look for users who have similar profiles as stated in the requirements detailed earlier in this document. In order to refine the search to include (for example) looking for agents who have a motivation to teach, we will need to develop the specification to define ways in which agents can broadcast that they are able to teach certain plans. This will come in later versions of this specification.

4 Concluding remarks

No one could have predicted the rise in technologies for facilitating different kinds of online social behaviour. Despite a sometimes limited scope of interact possibilities (such as liking, or rating content), huge numbers of us choose to socialise in this way. More and more technology platforms are being released, aiming to encourage us to spend our social time on them. Not only that, but we are now seeing a whole range of such systems that encourage us to spend our learning time on them, making use of a range of techniques to allow the learning experience to be less isolated and more social, particularly around the idea of peer feedback and assessment.

Given this explosion of systems for social experiences including social learning experiences, it is perhaps a little surprising that the multi-agent systems (MAS) community, with all its rich work on agency, coordination, norms and regulated social behaviour has not been more involved in taking up the challenge of trying to understand the science of such systems and in turn bringing that understanding into well-defined methodologies for designing compelling systems.

In this paper, we have shown that it is possible to use a standard agent-based formal specification methodology to model various aspects of a social learning environment. Building on that we have shown how such an architecture can be used to solve problems in these environments, such as selecting learning plans and selecting other users of interest. In parallel to this formal modelling work, we are building real social learning environments and trialling them at scale in our own institution and beyond, as part of a research programme investigating social learning. Now we have systems with users and data, we are investigating how our agent concepts can be operationalised to solve real problems within our systems. Our work is significant because we are bridging the theory/ practice divide.

Relating the theory and practice of sociological agent systems to the design of socio-technical systems more generally also enables us in future work to consider a range of questions about how the scientific social multi-agent approach that the MAS has developed for 25 years or more can be applied to the analysis and design of systems such as ours. Questions that quickly present themselves are: could we start to map out the space of such systems relating technology to sociality in a useful way? Could we start to provide platforms and design methodologies for building such systems in the future using a regulated MAS approach? Indeed these are some of the questions we are investigating with partners on our research project.

This paper is our first foray into these woods in describing an agent-based approach to the design of a community of human and learning agents working in the common interests of learning how to play musical instruments together. We hope that we will increasingly see the huge body of work that has been developed in our community over the last 25 years or so become mainstream in the analysis, design and specification of future instances of such systems.

Acknowledgments

The work reported in this paper is part of the PRAISE (Practice and Performance Analysis Inspiring Social Education) project which is funded under the EU FP7 Technology Enhanced Learning programme, grant agreement number 318770.

References

[Baylor and Group, 2003] Amy L Baylor and PALS (Pedagogical Agent Learning Systems) Research Group. The impact of three pedagogical agent roles. In *Proceedings of the second international joint conference on Autonomous*

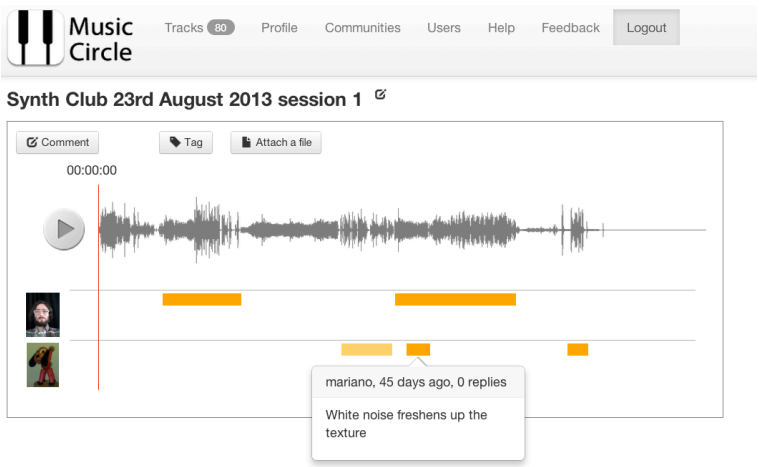


Figure 1: The music discussion user interface

agents and multiagent systems, AAMAS '03, pages 928–929, New York, NY, USA, 2003. ACM.

[D'Inverno and Luck, 2003] Mark D'Inverno and Michael Luck. *Understanding agent systems*. Springer, 2003.

[Johnson *et al.*, 2000] W. Lewis Johnson, Jeff W. Rickel, and James C. Lester. Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of . . .*, pages 47–78, 2000.

[Kim and Baylor, 2006] Yanghee Kim and AL Baylor. Pedagogical agents as learning companions: The role of agent competency and type of interaction. *Educational Technology Research and Development*, 54(3):223–243, 2006.

[Koller and Ng,] Daphne Koller and Andrew Ng. The Online Revolution : Education at Scale. Technical report, Stanford University.

[Lester *et al.*, 1997] JC Lester, SA Converse, and SE Kahler. The persona effect: affective impact of animated pedagogical agents. In *CHI 97 Conference on Human Factors in Computing Systems*, Atlanta, 1997.

[Luck and D'Inverno, 1995] Michael Luck and Mark D'Inverno. A formal framework for agency and autonomy. *Proceedings of the first international conference on Multi-Agent Systems*, 254260, 1995.

[Magnus *et al.*, 2010] H Magnus, S Annika, and S Björn. Building a Social Conversational Pedagogical Agent-Design Challenges and Methodological Approaches. In Diana Perez-Marin and Ismael Pascual-Nieto, editors, *Diana Perez-Marin (Editor), Ismael Pascual-Nieto (Editor)*, pages 128–155. IGI Global, 2010.

[Pappano, 2012] Laura Pappano. The year of the MOOC. *The New York Times*, 2(12):2012, 2012.

[Quirino *et al.*, 2009] E Quirino, F Paraguaçu, and B Jacinto. SSDCVA: Support System to the Diagnostic of Cerebral Vascular Accident For Physiotherapy Students. In *22nd IEEE International Symposium on Computer-Based Medical Systems, CBMS*, pages 2–5, 2009.

[Rodriguez, 2012] O Rodriguez. MOOCs and the AI-Stanford like Courses: two successful and distinct course formats for massive open online courses. *European Journal of Open, Distance, and E-Learning*, 2012.

[Sklar and Richards, 2006] Elizabeth Sklar and Debbie Richards. The use of agents in human learning systems. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, pages 767–774, New York, NY, USA, 2006. ACM.

[Soliman and Guetl, 2010] M Soliman and Christian Guetl. Intelligent pedagogical agents in immersive virtual learning environments: A review. In *MIPRO, 2010 Proceedings of the 33rd International Convention*. IEEE Computer Society Press, 2010.

[Spoelstra and Sklar, 2007] Maartje Spoelstra and Elizabeth Sklar. Using simulation to model and understand group learning. In *Proc. AAMAS'07 Workshop on Agent Based Systems for Human Learning and Entertainment*, 2007.

[Tao *et al.*, 2009] Xuehong Tao, YL Theng, and Nicola Yelland. Learning through argumentation with cognitive virtual companions. In C Fulford and George Siemens, editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications3179-3194*, pages 3179–3194, 2009.

[Xiao *et al.*, 2004] Jun Xiao, John Stasko, and Richard Catrambone. An Empirical Study of the Effect of Agent Competence on User Performance and Perception. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 178–185, 2004.

Implementing feedback in creative systems: A workshop approach

Joseph Corneli[1] and Anna Jordanous[2]
[1] Department of Computing, Goldsmiths College, University of London
[2] School of Computing, University of Kent

Abstract

One particular challenge in AI is the computational modelling and simulation of creativity. Feedback and learning from experience are key aspects of the creative process. Here we investigate how we could implement feedback in creative systems using a social model. From the field of creative writing we borrow the concept of a Writers Workshop as a model for learning through feedback. The Writers Workshop encourages examination, discussion and debates of a piece of creative work using a prescribed format of activities. We propose a computational model of the Writers Workshop as a roadmap for incorporation of feedback in artificial creativity systems. We argue that the Writers Workshop setting describes the anatomy of the creative process. We support our claim with a case study that describes how to implement the Writers Workshop model in a computational creativity system. We present this work using patterns other people can follow to implement similar designs in their own systems. We conclude by discussing the broader relevance of this model to other aspects of AI.

1 Introduction

In educational applications it would be useful to have an automated tutor that can read student work and make suggestions based on diagnostics, like, is the paper wrong, and if so how? What background material should be recommended to the student for review?

In the current paper, we "flip the script" and look at what we believe to be a more fundamental problem for AI: computer programs that can themselves learn from feedback. After all, if it was easy to build great automatic tutors, they would be a part of everyday life. As potential users (thinking from both sides of the desk) we look forward to a future when that is the case.

Along with automatic tutoring, computational creativity is a challenge within artificial intelligence where feedback plays a vital part (for example Pérez y Pérez, Aguilar, & Negrete, 2010; Pease, Guhe, & Smaill, 2010). Creativity cannot happen in a 'silo' but instead is influenced and affected by feedback and interaction with others (Csikszentmihalyi, 1988; Saunders, 2012). Computational creativity researchers are starting to place more emphasis on social interaction and feedback in their systems and models (Saunders, 2012; Gervás & León, 2014; Corneli et al., 2015). Still, nearly 3 in 4 papers at the 2014 International Conference for Computational Creativity[1] failed to acknowledge the role of feedback or social communication in their computational work on creativity.

To highlight and contribute towards modelling feedback as a crucial part of creativity, we propose in this paper a model of computational feedback for creative systems based on Writers Workshops (Gabriel, 2002), a literary collaborative practice that encourages interactive feedback within the creative process. We introduce the Writers Workshop concept (Section 2) and critically reflect on how it could encourage serendipity and emergence in computational models of intelligence and creativity. These considerations lead us to propose a Writers Workshop computational model of feedback in computational creativity and AI systems (Section 2.1), the central contribution of this paper. In Section 3 we consider how the Writers Workshop model fits into previous work in various related areas. While we acknowledge that this paper is offering a roadmap for this model rather than a full implementation, we consider how the model could be practically implemented in a computational system and report our initial implementation work (Section 4). In concluding discussions, we reflect on divergent directions in which this work could potentially be useful in the future.

2 The Writers Workshop

Richard Gabriel (2002) describes the practise of Writers Workshops that has been put to use for over a decade within the Pattern Languages of Programming (PLoP) community. The basic style of collaboration originated much earlier with groups of literary authors who engage in peer-group critique. Some literary workshops are open as to genre, and happy to accommodate beginners, like the Minneapolis Writers Workshop[2]; others are focused on professionals working within a specific genre, like the Milford Writers Workshop.[3]

[1] ICCC is the key international conference for research in computational creativity.

[2] `http://mnwriters.org/how-the-game-works/`

[3] `http://www.milfordsf.co.uk/about.htm`

The practices that Gabriel describes are fairly typical:

- Authors come with work ready to present, and read a short sample.
- This work is generally work in progress (and workshopping is meant to help improve it). Importantly, it can be early stage work. Rather than presenting a created artefact only, activities in the workshop can be aspects of the creative process itself. Indeed, the model we present here is less concerned with after-the-fact assessment than it is with dealing with the formative feedback that is a necessary support for creative work.
- The sample work is then discussed and constructively critiqued by attendees. Presenting authors are not permitted to rebut these comments. The commentators generally summarise the work and say what they have gotten out of it, discuss what worked well in the piece, and talk about how it could be improved.
- The author listens and may take notes; at the end, he or she can then ask questions for clarification.
- Generally, non-authors are either not permitted to attend, or are asked to stay silent through the workshop, and perhaps sit separately from the participating authors/reviewers.[4]

Essentially, the Writers Workshop is somewhat like an interactive peer review. The underlying concept is reminiscent of Bourdieu's *fields of cultural production* (Bourdieu, 1993) where cultural value is attributed through interactions in a community of cultural producers active within that field.

2.1 Writers Workshop as a computational model

The use of Writers Workshop in computational contexts is not an entirely new concept. In PLoP workshops, authors present design patterns and pattern languages, or papers about patterns, rather than more traditional literary forms like poems, stories, or chapters from novels. Papers must be workshopped at a PLoP or EuroPLoP conference in order to be considered for the *Transactions on Pattern Languages of Programming* journal. A discussion of writers workshops in the language of design patterns is presented by Coplien and Woolf (1997).

The steps in the workshop can be distilled into the following phases, each of which could be realised as a separate computational step in an agent-based model:

1. Author: `presentation`
2. Critic: `listening`
3. Critic: `feedback`
4. Author: `questions`
5. Critic: `replies`
6. Author: `reflections`

[4]Here we present Writers Workshops as they currently exist; however this last point is debatable. Whether non-authors should be able to participate or not is an interesting avenue for experimentation both in human and computational contexts. The workshop dialogue itself may be considered an "art form" whose "public" may potentially wish to consume it in non-participatory ways. Compare the classical Japanese *renga* form (Jin'Ichi, Brazell, & Cook, 1975).

The `feedback` step may be further decomposed into `observations` and `suggestions`. This protocol is what we have in mind in the following discussion of the Writers Workshop.[5]

Dialogue example

Note that for the following dialogue to be possible computationally, it would presumably have to be conducted within a lightweight process language. Nevertheless, for convenience, the discussion will be presented here as if it was conducted in natural language. Whether contemporary systems have adequate natural language understanding to have interesting interactions is one of the key unanswered questions of this approach, but protocols such as the one described above are sufficient to make the experiment.

For example, here's what might happen in a discussion of the first few lines of a poem, "On Being Malevolent". As befitting the AI-theme of this workshop, "On Being Malevolent" is a poem written by an early user-defined flow chart in the FloWr system (known at the time as Flow) (Colton & Charnley, 2014).

> FLOW: "*I hear the souls of the damned waiting in hell. / I feel a malevolent spectre hovering just behind me / It must be his birthday*."
>
> SYSTEM A: I think the third line detracts from the spooky effect, I don't see why it's included.
>
> SYSTEM B: It's meant to be humourous – in fact it reminds me of the poem you presented yesterday.
>
> MODERATOR: Let's discuss one poem at a time.

Even if, perhaps and especially because, "cross-talk" about different poems bends the rules, the dialogue could prompt a range of reflections and reactions. System A may object that it had a fair point that has not been given sufficient attention, while System B may wonder how to communicate the idea it came up with without making reference to another poem. Here's how the discussion given as example in Section 2 might continue, if the systems go on to examine the next few lines of the poem.

> FLOW: "*Is God willing to prevent evil, but not able? / Then he is not omnipotent / Is he able, but not willing? / Then he is malevolent.*"
>
> SYSTEM A: These lines are interesting, but they sound a bit like you're working from a template, or like you're quoting from something else.
>
> SYSTEM B: Maybe try an analogy? For example, you mentioned birthdays: you could consider an analogy to the conflicted feelings of someone who knows in advance about her surprise birthday party.

[5]The connections between Writers Workshops and design patterns, noted above, appear to be quite natural, in that the steps in the workshop protocol roughly parallel the typical components of design pattern templates: *context*, *problem*, *solution*, *rationale*, *resolution of forces*.

Figure 1: A paper prototype for applying the *Successful Error* pattern following a workshop-like sequence of steps

This portion of the discussion shifts the focus of the discussion onto a line that was previously considered to be spurious, and looks at what would happen if that line was used as a central metaphor in the poem.

> FLOW: Thank you for your feedback. My only question is, System B, how did you come up with that analogy? It’s quite clever.
> SYSTEM B: I’ve just emailed you the code.

Whereas the systems were initially reviewing poetry, they have now made a partial genre shift, and are sharing and remixing code. Such a shift helps to get at the real interests of the systems (and their developers). Indeed, the workshop session might have gone better if the systems had focused on exchanging and discussing more formal objects throughout.

2.2 How the Writers Workshop can lead to computational serendipity

Learning involves engaging with the unknown, unfamiliar, or unexpected and synthesising new understanding (Deleuze, 2004 [1968]). In the workshop setting, learning can develop in a number of unexpected ways, and participating systems need to be prepared for this. One way to evaluate the idea of a Writers Workshop is to ask whether it can support learning that is in some sense *serendiptious*, in other words, whether it can support discovery and creative invention that we simply couldn’t plan for or orchestrate in another way.

Figure 1 shows a paper prototype showing how one of the “patterns of serendipity” that were collected by Van Andel (1994) might be modelled in a workshop-like dialogue sequence. The patterns also help identify opportunities for serendipity at several key steps in the workshop sequence.

Serendipity Pattern: *Successful error*. Van Andel describes the creation of Post-it™ Notes at 3M. One of the instrumental steps was a series of internal seminars in which 3M employee Spencer Silver described an invention he was sure was interesting, but was unsure how to turn into a useful product: weak glue. The key prototype that came years later was a sticky bookmark, created by Arthur Fry. In the Writers Workshop, authors similarly have the opportunity to share things that they find interesting, but that they are not certain about. The author may want to ask a specific question about their creation: Does x work better than y? They may flag certain parts of the work as especially problematic. They may think that a certain portion of the text is interesting or important, without being sure why. Although there is no guarantee that a participating critic will be able to take these matters forward, sometimes they do – and the workshop environment will produce something that the author wouldn’t have thought of alone.

Serendipity Pattern: *Outsider*. Another example from van Andel considers the case of a mother whose son was aflicted by a congenital cateract, who suggested to her doctor that rubella during pregnancy may have been the cause. In the workshop setting, someone who is not an “expert” may come up with a sensible idea or suggestion based on their own prior experience. Indeed, these suggestions may be more sensible than the ideas of the author, who may be to close to the work to notice radical improvements.

Serendipity Pattern: *Wrong hypothesis*. A third example describes the discovery that lithium can have a therapeutic effect in cases of mania. Originally, lithium carbonate had merely been used a control by John Cade, who was interested in the effect of effect of uric acid, present in soluble

lithium urate. Cade was searching for causal factors in mania, not therapies for the condition: but he found that lithium carbonate had an unexpected calming effect. Similarly, in the workshop, the author may think that a given aspect of their creation is the interesting "active ingredient," and it may turn out that another aspect of the work is more interesting to critics. Relatedly, the author may not fully comprehend a critic's feedback and may have to ask follow-up questions to understand it.

Serendipity Pattern: ***Side effect.*** A fourth example described by van Andel concerns Ernest Huant's discovery that nicotinamide, which he used to treat side-effects of radiation therapy, also proved efficacious against tuberculosis. In the workshop setting, one of the most important places where a side-effect may occur concerns feedback from the critic to the author. In the simple case, feedback may trigger revisions to the work under discussion. In a more general, and more unpredictable case, feedback may trigger broader revisions to the generative codebase.

This collection of patterns shows the likelihood of unexpected results coming out of the communication between author and critics. This suggests several guidelines for system development, which we will discussed in a later section.

Further guidelines for structuring and participating in traditional writers workshops are presented by Linda Elkin in (Gabriel, 2002, pp. 201–203). It is not at all clear that the same ground rules should apply to computer systems. For example, one of Elkin's rules is that "Quips, jokes, or sarcastic comments, even if kindly meant, are inappropriate." Rather than forbidding humour, it may be better for individual comments to be rated as helpful or non-helpful. Again, in the first instance, usefulness and interest might be judged in terms of explicit criteria for serendipity; see (Corneli, Pease, Colton, Jordanous, & Guckelsberger, 2014; Pease, Colton, Ramezani, Charnley, & Reed, 2013). The key criterion in this regard is the *focus shift*. This is the creation of a novel problem, comprising the move from discovery of interesting data to the invention of an application. This process is distinct from identifying routine errors in a written work. Nevertheless, from a computational standpoint, noticing and being robust to certain kinds of errors is often a preliminary step. For example, the work might contain a typo, grammatical or semantic error, while being logically sound. In a programming setting, this sort of problem can lead to crashing code, or silent failure. In general communicative context, argumentation may be logically sound, but not practically useful or poorly exposited. Finally, even a masterful, correct, and fully spellchecked piece of argumentation may not invite further dialogue, and so may fail to open itself to further learning. Identifying and engaging with this sort of deeper issue is something that skillful workshop participants may be able to do. Dialogue in the workshop can build on strong or less strong work – but provoking interpretative thoughts and comments always require a thoughtful critical presence and the ability to engage. This can be difficult for humans and poses a range of challenges for computers – but also promises some interesting results.

3 Related work

In considering the potential and contribution of the Writers Workshop model outlined in Section 2, we posit that the Writers Workshop model is useful for encouraging feedback in computational systems, and in particular systems that are designed to be creative or serendipitous.

Feedback has long been a central concept in AI-related fields such as cybernetics (Ashby, 1956; Seth, 2015). Feedback about feedback (and &c for higher orders) is understood to be relevant to thinking about *learning* and *communication* (Bateson, 1972). We now consider the importance of the roles that communicative feedback play in computational creativity and computational serendipity and discuss previous related work in incorporating feedback into such computational systems.

3.1 Feedback in computational creativity

Creativity is often envisaged as involving cyclical processes (e.g. Dickie's (1984) art circle, Pease and Colton's (2011) Iterative Development-Expression-Appreciation model). There are opportunities for embedded feedback at each step, and the creative process itself is "akin to" a feedback loop. However, despite these strong intimations of the central importance of feedback in the creative process, our sense is that feedback has not been given a central place in research on computational creativity. In particular, current systems in computational creativity, almost as a rule, do *not* consume or evaluate the work of other systems.[6]

Gervás and León (2014) theorise a creative cycle of narrative development as involving a Composer and an Interpreter, in such a way that the Composer has internalised the interpretation functionality. Individual creativity is not the poor relation of social creativity, but its mirror image. Nevertheless, even when computer models explicitly involve multiple agents and simulate social creativity (like Saunders & Gero, 2001), they rarely make the jump to involve multiple systems. The "air gap" between computationally creative systems is very different from the historical situation in human creativity, in which different creators and indeed different cultural domains interact vigorously (Geertz, 1973).

3.2 Feedback in computational serendipity

The term computational serendipity is rather new, but its foundations are well established in prior research.

Grace and Maher (2014) examine *surprise* in computing, seeking to "adopt methods from the field of computational creativity [...] to the generation of scientific hypotheses." This is an example of an effort focused on computational *invention*.

An area of AI where serendipity can be argued to play an important part is in pattern matching. Current computer programs are able to identify known patterns and "close matches" in data sets from certain domains, like music (Meredith, Lemström, & Wiggins, 2002). Identifying known

[6] An exception to the rule is Mike Cook's AppreciationBot (`https://twitter.com/AppreciationBot`), which is a reactive automaton that "appreciates" tweets from MuseumBot.

patterns is a special case of the more general concept of *pattern mining* (Bergeron & Conklin, 2007). In particular, the ability to extract *new* higher order patterns that describe exceptions is an example of "learning from feedback." Deep learning and evolutionary models increasingly use this sort of idea to facilitate strategic discovery (Samothrakis & Lucas, 2011). Similar ideas are considered in business applications under the heading "process mining" (Van Der Aalst, 2011).

In earlier work (Corneli et al., 2014, 2015), we used the idea of dialogue in a Writers Workshop framework to sketch a "theory of poetics rooted in the making of boundary-crossing objects and processes" and described (at a schematic level) "a system that can (sometimes) make 'highly serendipitous' creative advances in computer poetry" while "drawing attention to theoretical questions related to program design in an autonomous programming context."

3.3 Communications and feedback

The Writers Workshop heavily relies on communication of feedback within the workshop. Gordon Pask's conversation theory, reviewed in (Pask, 1984; Boyd, 2004), goes considerably beyond the simple process language of the workshop, although there are structural parallels. We see that a basic Pask-style learning conversation bears many similarities to the Writers Workshop model of communicative feedback (Boyd, 2004, p. 190):

1. Conversational participants are carrying out some actions and observations;
2. Naming and recording what action is being done;
3. Asking and explaining why it works the way it does;
4. Carrying out higher-order methodological discussion; and,
5. Trying to figure out why unexpected results occured.

Variations to the underlying system, protocol, and the schedule of events should be considered depending on the needs and interests of participants, and several variants can be tried. On a pragmatic basis, if the workshop proved quite useful to participants, it could be revised to run monthly, weekly, or continuously.[7]

4 Case study: Flowcharts and Feedback

This section describes work that is currently underway to implement the Writers Workshop model, not only within one system but as a new paradigm for collaboration among disparate projects. In order to bring in other participants, we need a neutral environment that is not hard to develop for: the FloWr system mentioned in Section 2.1 offers one such possibility. The basic primary objects in the FloWr system are *flowcharts*, which are comprised of interconnected *process nodes* (Charnley, Colton, & Llano, 2014; Colton & Charnley, 2014). Process nodes specify input and output types, and internal processing can be implemented in Java, or other languages that interoperate with the JVM, or by invoking external web services. One of the common applications to date is to generate computer poetry, and we will focus on that domain here.

A basic set of questions, relative to this system's components, are as follow:

1. *Population of nodes*: What can they do? What do we learn when a new node is added?
2. *Population of flowcharts*: Pease et al. (2013) have described the potentially-serendipitous repair of "broken" flowcharts when new nodes become available; this suggests the need for test-driven development framework.
3. *Population of output texts*: How to assess and comment on a generated poetic artefact?

In a further evolution of the system, the sequence of steps in a Writers Workshop could itself be spelled out as a flowchart. The process of reading a poem could be conceptualised as generating a semantic graph (Harrington & Clark, 2007; Francisco & Gervás, 2006). Feedback could be modelled as annotations to a text, including suggested edits. These markup directives could themselves be expressed as flowcharts. A standardised set of markup structures may partially obviate the need for strong natural language understanding, at least in interagent communication. Thus, we could agree that `observations` will consist of stand-off annotations that connect textual passages to public URIs using a limited comparison vocabulary, and `suggestions` will consist of simple stand-off line-edits, which may themselves be marked up with rationale. These restrictions, and similar restrictions around constrained turn-taking, could be progressively widened in future versions of the system. The way the poems that are generated, the models of poems that are created, and the way the feedback is generated, all depend on the contributing system's body of code and prior experience, which may vary widely between participating systems. In the list of functional steps below, all of the functions could have a subscripted "$\mathcal{E}$", which is omitted throughout. Exchanging path dependent points of view will tend to produce results that are different from what the individual participating systems would have come up with on their own.

I. Both the author and critic should be able to work with a model of the text. Some of the text's features may be explicitly tagged as "interesting." Outstanding questions may possibly be brought to the attention of critical listeners, e.g. with the request to compare two different versions of the poem (`presentation`, `listening`).
   1. *A model of the text.* $m : T \to M$.
   2. *Tagging elements of interest.* $\mu : M \to I$.

II. Drawing on its experience, the critic will use its model of the poem to formulate feedback (`feedback`).
   1. *Generating feedback.* $f : (T, M, I) \to F$.

[7] For a comparison case in computer Go, see `http://cgos.computergo.org/`.

III. Given the constrained framework for feedback, statements about the text will be straightforward to understand, but rationale for making these statements may be more involved (`questions`, `replies`).

   1. *Asking for more information.* $q : (M, F, I) \rightarrow Q$.
   2. *Generating rationale.* $a : (M, F, Q) \rightarrow \Delta F$.

IV. Finally, feedback may affect the author's model of the world, and the way future poems are generated (`reflection`).

   1. *Updating point of view.* $\rho : (M, F) \rightarrow \Delta \mathcal{E}$.

The final step is perhaps the most interesting one, since it invites us to consider how individual elements of feedback can "snowball" and go beyond line-edits to a specific poem to much more fundamental changes in the way the presenting agent writes poetry. Here methods for pattern mining, discussed in Section 3.2, are particularly relevant. If systems can share code (as in our sample dialogue in Section 2.1) this will help with the rationale-generating step, and may also facilitate direct updates to the codebase. However, shared code may be more suitably placed into the common pool of resources available to FloWr than copied over as new "intrinsic" features of an agent.

Although different systems with different approaches and histories are important for producing unexpected effects, "offline" programmatic access to a shared pool of nodes and existing flowcharts may be useful. Outside of the workshop itself, agents may work to recombine nodes based on their input and output properties to assemble new flowcharts. This can potentially help evaluate and evolve the population of nodes programmatically, if we can use this sort of feedback to define fitness functions. The role of temporality is interesting: if the workshop takes place in real time, this will require different approaches to composition that takes place offline (Perez, Samothrakis, Lucas, & Rohlfshagen, 2013). Complementing these "macro-level" considerations, it is also worth commenting on the potential role of "micro-level" feedback within flowcharts. Local evaluation of output from a predecessor node could feed backwards through the flowchart, similar to backpropagation in neural networks. This would rely on a reduced version of the functional schema described above.

5 Concluding discussion and future directions

We have described a *general* and *computationally feasible* model for using feedback in AI systems, particularly creative systems. The Writers Workshop concept, borrowed from creative writing, is transformed into a model of a structured approach to eliciting, processing and learning from feedback. To better evaluate how the Writers Workshop model helps us advance in our goal of incorporating feedback into artificial creativity, we critically considered how the model fits into related work. In particular, we found that serendipity, a key concept within creativity and AI more generally, is a concept with which the Writers Workshop model could assist computational progress. In this respect, we should highlight the difference between "global" analytics describing the collection of nodes and flowcharts in the FloWr ecosystem, and the path-dependent process of analysis and synthesis that takes place in a workshop setting.

Our preliminary implementation work (Section 4) shows that the model can be transfered to a functional implementation. This work highlights several considerations relevant to further work with the Writers Workshop model:

- Each contributing system should come to the workshop with at least a basic awareness of the workshop protocol, with work to share, and prepared to give constructive feedback to other systems.
- The workshop itself needs to be prepared, with a suitable communication platform and a moderator or global flowchart for moving the discussion from step to step.
- A controlled vocabulary for communications and interaction would be a worthwhile pursuit of future research, perhaps based on an ontology inspired by the Interaction Network Ontology.[8]
- In order to get the most value out of the workshop experience, systems (and their wranglers) should ideally have questions they are investigating. As discussed above, prior experience plays an important role in every step. This opens up a range of issues for further research on modeling motivations and learning from experience.
- Systems should be prepared to give feedback, and to carry out evaluations of the helpfulness (or not) of feedback from other systems and of the experience overall.

Developing systems that could successfully navigate this collaborative exercise would be a significant advance in the field of computational creativity. Since the experience is about learning rather than winning, there is little motivation to "game the system" (cf. Lenat, 1983). Instead the emphasis is squarely upon mutual benefit: computational systems helping to develop each other through communication and feedback.

The benefits of the Writers Workshop approach could innovate well beyond models for feedback and communication within a particular environment or restricted domain. Following the example of the Pattern Languages of Programming (PLoP) community, we propose that the Writers Workshop model could be deployed within the Computational Creativity community to design a workshop in which the participants are computer systems instead of human authors. The annual International Conference on Computational Creativity (ICCC), now entering its sixth year, could be a suitable venue.

Rather than the system's creator presenting the system in a traditional slideshow and discussion, or a system "Show and Tell," the systems would be brought to the workshop and would present their own work to an audience of other systems, in a Writers Workshop format. This could be accompanied by a short paper for the conference proceedings written

[8] The Interaction Network Ontology primarily describes interactions within humans as opposed to within human societies; a distinct *Social* Interaction Ontology does not seem to exist at present. However, the classes of the Interaction Network Ontology appear to be quite broadly relevant. This ontology is documented at `http://www.ontobee.org/browser/index.php?o=INO`. Its URI is `http://svn.code.sf.net/p/ino/code/trunk/src/ontology/INO.owl`.

by the system's designer describing the system's current capabilities and goals. If the Workshop really works well, future publications might adapt to include traces of Workshop interactions, commentary from a system on other systems, and offline reflections on what the system might change about its own work based on the feedback it receives. Paralleling the PLoP community, it could become standard to incorporate the workshop into the process of peer review for the new *Journal of Computational Creativity*.[9] AI systems that review each other would surely be a major demonstration and acknowledgement of the usefulness of feedback within AI.

In closing, we wish to return briefly to the scenario of computer generated feedback in educational contexts that we raised at the beginning of this paper and then set aside. The elements of our functional design for sharing feedback among computational agents has a range of features that continue to be relevant for generating useful feedback with human learners. Students are experience-bound, and a robust approach to formative assessment and feedback should take into account the student's historical experience, so far as this can be known or inferred. In order for feedback, recommendations, and so on to adequately take individual history into account, sophisticated modelling and reasoning would be required. Nevertheless, from the point of view of participating computational agents, a student may simply look like another agent. It is in this regard that computational models of learning from feedback are seen as fundamental.

Acknowledgement

Joseph Corneli's work on this paper was supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number 611553 (COINVENT).

References

Ashby, W. R. (1956). *An introduction to cybernetics.* London, UK: Chapman & Hail Ltd.

Bateson, G. (1972). *Steps to an ecology of mind*. Chicago: University of Chicago Press.

Bergeron, M., & Conklin, D. (2007). Representation and discovery of feature set patterns in music. In *International Workshop on Artificial Intelligence and Music, at IJCAI-07, The Twentieth International Joint Conference on Artificial Intelligence* (pp. 1–12).

Bourdieu, P. (1993). *The field of cultural production: Essays on art and literature*. Cambridge, UK: Polity Press.

Boyd, G. M. (2004). Conversation theory. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (2nd ed., pp. 179–197). Lawrence Erlbaum.

Charnley, J., Colton, S., & Llano, M. T. (2014). The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the 5th International Conference on Computational Creativity.*

Colton, S., & Charnley, J. (2014). Towards a Flowcharting System for Automated Process Invention. In D. Ventura, S. Colton, N. Lavrac, & M. Cook (Eds.), *Proceedings of the Fifth International Conference on Computational Creativity.*

Coplien, J. O., & Woolf, B. (1997). A pattern language for writers' workshops. *C++ report*, *9*, 51–60.

Corneli, J., Jordanous, A., Shepperd, R., Llano, M. T., Misztal, J., Colton, S., & Guckelsberger, C. (2015). Computational Poetry Workshop: Making Sense of Work in Progress. In *Proceedings of the Sixth International Conference on Computational Creativity.* Retrieved from `http://metameso.org/~joe/docs/poetryICCC-wip.pdf`

Corneli, J., Pease, A., Colton, S., Jordanous, A., & Guckelsberger, C. (2014). *Modelling serendipity in a computational context.* Retrieved from `http://arxiv.org/abs/1411.0440` (Under review.)

Csikszentmihalyi, M. (1988). Society, culture, and person: a systems view of creativity. In R. J. Sternberg (Ed.), *The nature of creativity* (chap. 13). Cambridge, UK: Cambridge University Press.

Deleuze, G. (2004 [1968]). *Difference and repetition*. Bloomsbury Academic.

Dickie, G. (1984). *The art circle: A theory of art*. Haven.

Francisco, V., & Gervás, P. (2006). Automated mark up of affective information in English texts. In *Text, speech and dialogue* (pp. 375–382).

Gabriel, R. P. (2002). *Writer's Workshops and the Work of Making Things: Patterns, Poetry...* Addison-Wesley Longman Publishing Co., Inc.

Geertz, C. (1973). *The interpretation of cultures: Selected essays*. Basic Books (AZ).

Gervás, P., & León, C. (2014). Reading and Writing as a Creative Cycle: The Need for a Computational Model. In *5th International Conference on Computational Creativity, ICCC 2014.* Ljubljana, Slovenia.

Grace, K., & Maher, M. L. (2014). Using Computational Creativity to Guide Data-Intensive Scientific Discovery. In Y. Gil & H. Hirsh (Eds.), *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence.* (Discovery Informatics Workshop: Science Challenges for Intelligent Systems.)

Harrington, B., & Clark, S. (2007). ASKNet: Automated Semantic Knowledge Network. In A. Howe & R. Holte (Eds.), *Procs. of the Twenty-Second AAAI Conference on Artificial Intelligence* (Vol. 2, pp. 889–895). AAAI Press.

Jin'Ichi, K., Brazell, K., & Cook, L. (1975). The Art of Renga. *Journal of Japanese Studies*, 29–61.

Lenat, D. B. (1983). EURISKO: a program that learns new heuristics and domain concepts: the nature of heuristics III: program design and results. *Artificial Intelligence*, *21*(1), 61–98.

Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, *31*(4), 321–345.

[9] `http://www.journalofcomputationalcreativity.cc`

Pask, G. (1984). Review of conversation theory and a protologic (or protolanguage), Lp. *ECTJ*, *32*(1), 3-40. Retrieved from http://dx.doi.org/10.1007/BF02768767 doi: 10.1007/BF02768767

Pease, A., & Colton, S. (2011). Computational creativity theory: Inspirations behind the FACE and the IDEA models. In *Proceedings of the Second International Conference on Computational Creativity.*

Pease, A., Colton, S., Ramezani, R., Charnley, J., & Reed, K. (2013). A Discussion on Serendipity in Creative Systems. In *Proceedings of the Fourth International Conference on Computational Creativity.*

Pease, A., Guhe, M., & Smaill, A. (2010). Some aspects of analogical reasoning in mathematical creativity. In *Proceedings of the International Conference on Computational Creativity* (p. 60-64). Lisbon, Portugal.

Perez, D., Samothrakis, S., Lucas, S., & Rohlfshagen, P. (2013). Rolling horizon evolution versus tree search for navigation in single-player real-time games. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation* (pp. 351–358).

Pérez y Pérez, R., Aguilar, A., & Negrete, S. (2010). The ERI-Designer: A computer model for the arrangement of furniture. *Minds and Machines*, *20*(4), 533-564.

Samothrakis, S., & Lucas, S. (2011). Approximating n-player Behavioural Strategy Nash Equilibria Using Coevolution. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 1107–1114).

Saunders, R. (2012). Towards autonomous creative systems: A computational approach. *Cognitive Computation*, *4*(3), 216–225.

Saunders, R., & Gero, J. S. (2001). The digital clockwork muse: A computational model of aesthetic evolution. In *Proc. Annual Convention of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (SSAISB)* (pp. 12–21).

Seth, A. (2015). The cybernetic bayesian brain: From interoceptive inference to sensorimotor contingencies. In T. Metzinger & J. Windt (Eds.), *Open mind project* (p. 1-24). Frankfurt: MIND Group.

Van Andel, P. (1994). Anatomy of the Unsought Finding. *The British Journal for the Philosophy of Science*, *45*(2), pp. 631–648.

Van Der Aalst, W. (2011). *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media.

The Comic Strip Game: observing the impact of implicit feedback in the content creation process

Fiammetta Ghedini and **Benjamin Frantz** and **François Pachet** and **Pierre Roy**
SONY CSL Paris
fiammettaghedini@gmail.com

Abstract

The Comic Strip Game is a system allowing users to create dialogues for speechless cartoon strips during shared, online content creation sessions. This paper describes the results of a protocol providing each participant with implicit feedback and inspiration from other participants. We observed the behaviour of subjects and investigate the impact of other participants' behaviour on their creative process.

1 Introduction

We posit that a creation task involves a succession of production and self-evaluation steps, which, usually, are not explicit and thus cannot be observed [Lubart 2013]. We have designed an experimental protocol in which production and evaluation steps are explicit, as well as the influence that other people work can have on one's own. Specifically, the objective of the Comic Strip Game is to investigate (1) positive or negative bias in implicit self-evaluation of creativity, (2) the impact of implicit feedback from other participants, (3) the attachment to one's own creation and (4) the existence of consistent content creation strategies and their distribution on the subjects' population.

In order to do so, we designed an online system proposing ten different cartoon strips (see all the proposed strips in Annex I, at the end of this document). A strip is a series of images separated from each other, each image representing a character or a scene happening in a visual unity called "panel". In each panel of the strips, one or two characters are talking, by means of the traditional symbol of the "balloons", which were left blank, for the participants to write up. The content creation task is therefore to invent a story for the strip and to express it via the text in the balloons. In the Comic Strip Game, this content creation task is neither solitary nor collaborative: it is rather concerned with mechanisms of implicit feedback from other participants, as detailed in the protocol in the section below.

1.1 Motivations and background

Evaluation, both external and "internal" (self-evaluation) is a central issue of any creative process. The importance of self-evaluation and its role on the creative process itself, nevertheless, have not been extensively studied, probably more because of the difficulty of defining and isolating it with more subtle means that administrating questionnaires. This is why we thought that designing a system able to provide implicit feedback and imposing several implicit self-evaluation steps may be a valid methodology in order to investigate the creative process.

In general, previous literature has discussed whether the potential for external evaluation can affect the creative process; the first study by [Amabile, 1979], confirmed by [Bartis et al, 1998] highlighted a decrement in creativity due to external evaluation. [Szymanski & Harkins, 1992] partly confirmed the harmful effect of external evaluation on creativity (but not on the performance itself) during the process of generating as many uses as possible for an object. [Silvia and Philips 2004] suggests that also self-evaluation reduces creativity (for tasks involving generating remote associates and finding unusual uses for objects).

From another point of view, as external assessment of creativity can be taken into account when judges reach an agreement, the validity itself of self-evaluation in creativity has been questioned. For instance [Kaufman et al 2010] compared self-reports of creativity in four artistic domains to experts' judgments: external and self-evaluations did not correlate. In a similar study applied specifically to music, [Priest 2006] compared students' self-assessments of musical compositions and experts' assessments. In this study too, there was no significant correlation between the judges' evaluations and the students' reports.

On the other hand, we do not refer to the theoretical framework of collaborative creativity, which presents specific characteristics, such as "idea talk", variance in contributions, roles artificial or spontaneous attribution [Freeman 2014] which are made impossible by the constraints imposed by our system, for which the feedback has to remain implicit.

Indeed, the design of our study is meant to investigate the choices and evaluations present in the creative process bypassing an explicit self-evaluation step, by presenting the subjects with implicit feedback from other participants. Implicit feedback that every subject will receive from co-participants should also deflect the social loafing phenomenon, which is the tendency for individuals to lower

their productivity when in a group (see [Simms 2014] for an extensive review and [Williams et al. 1981] for how potential evaluation decreases social loafing.

1.2 Methodology

The experiment takes place online, where four subjects are randomly assigned to create dialogues for the same strip. As explained above, each strip consists in a set of images made of three to four panels (See Annex I), in which balloons are left blank. Each subject can complete from one to eight different stories.

Before the content creation session begins, subjects are required to declare their age, gender, mother tongue and all the languages they are fluent in. We then form groups of four subjects who will participate for a single session. The groups are formed randomly but each member has in common the knowledge of at least one language (the one in which the dialogues are going to be written) and the fact that it is the first and only time they created content for that specific strip. Subjects assigned to the same strip are anonymous and cannot directly interact with each other. Before the session begins, subjects are instructed on the task they will perform by means of a tutorial, with a particular emphasis on the fact that the objective is not only to write but also to choose the best story (even if the subject is not the author of it). To reinforce this concept and motivate the subjects, we stated that one of the participants would be chosen randomly to win a SONY tablet, among the participants who would choose (thus not necessarily write) one of the best stories.

During each stage of the content session creation, the subjects are producing dialogues (text) for one panel. When all the subjects have written and submitted their first dialogue, their texts are proposed to all four participants, including themselves. Subjects are instructed to choose other participants' proposals if they evaluate they have a greater potential for further developing the strip, therefore subjects, at this stage, can decide to pursue the story by selecting their own text proposal or to switch to the proposal of someone else. The same protocol applies for the second, third and fourth step.

We refer to the behaviour of choosing another subject's' proposal as "switch", therefore to subjects performing it as "switchers". We refer to the subjects who choose to continue their own production as "pursuers".

The final step consists in asking subjects to choose the story they evaluate as the best one. As explained above, before the beginning of the experiment subjects were instructed to choose the best one, regardless of how much they contributed to it.

For the sake of clarity, since the protocol is quite complex, we describe hereafter the typical development of one session (dialogues and behaviours come from our actual dataset, but names are fictional).

An example session from the Comic Strip Game

Vincent, Paul, Francesca and Benoit participate to their first session. Paul is Dutch, Francesca is Italian, Vincent and Benoit are French, but they are all fluent in English. They do not personally know each other.

The four subjects state their gender, age and spoken languages before they begin to play. When all these data is entered, they visualise the strip which the system has randomly chosen for them (the only condition is that nobody among them has already created a dialogue for this strip). The strip depicts, in this case, a ginger cat in different positions.

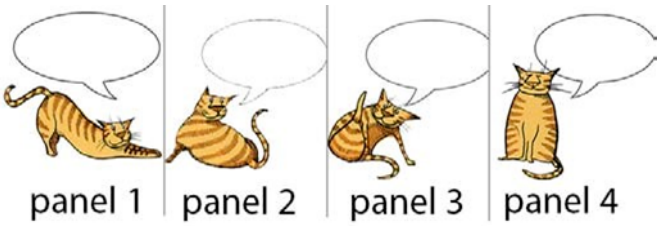


Figure 1: One of the proposed strips

They all propose a text for the first panel (See Fig.1), which is automatically inserted in the balloon.

Vincent proposes: Nothing like a good night
Paul proposes: Oh, c'est parfois difficile de se lever, le matin!
Francesca proposes: The Smooth Slope…
Benoit proposes: Usual stretching and we can start

They submit their proposal by clicking on a "send" button, and when everybody is done, they are able to read the proposal written by others. Now they can either choose to pursue their own story (that is to say, to write a text for Panel 2 by continuing with their own text) or to switch to the proposal of someone else. This is what happens:

Vincent chooses to "abandon" his text and switches to the text of Benoit: this means that during the next step, he will have to write up the second panel, as a continuation of "Usual stretching and we can start".

Paul, who probably got confused with the experiment's language as he has been writing in French, makes the same choice as Vincent: he switches in favour of Benoit's text.

Francesca keeps her text.

Benoit decides to switch and choses Vincent's text. Now they propose a text for the second panel (the first panel's text in italics):

Vincent proposes: *(1) Usual stretching and we can start* (2) It's important to be well prepared
Paul proposes: *(1) Usual stretching and we can start)* (2) In this position, I look quite fat!
Francesca proposes: *(1) The Smooth Slope* (2)…the Easter Egg…

Benoit proposes: *(1) Nothing like a good night* (2) Sorry for the back.
Now the subjects choose again the chain of two panels to be continued. It is worth mentioning that during this step (before proposing a text for the third panel) the subjects are exposed to the first implicit feedback from other participants, because they realise whether their first proposed text has been chosen by other subjects (and therefore its creative potential has been implicitly evaluated as good) or not. For instance, now Benoit, who abandoned his own proposal, realises that his text has been chosen by two other participants, Vincent and Paul. Vincent, once again, switches in favour of Benoit. He does not know who is submitting the text, as participants are invisible to each other and do not even have a nickname, so there is no possible bias in choosing a specific author: what is chosen is always and only the text. Paul switches in favour of the story of Francesca. He likes her idea (which he had probably not understood at the beginning): the cat is doing yoga. Francesca keeps her text. Benoit, this time, decides to keep his text.
Now the subjects propose a text for the third panel (in italics the first and second panel):
Vincent proposes: (1) Nothing like a good night) (2) Sorry for the back (3) Wow. I didn't know I could do that with my leg!
Paul proposes: (1) The Smooth Slope… (2)…the Easter Egg… (3) The One Beer for me…
Francesca proposes: (1) The Smooth Slope… (2)…the Easter Egg… (3) …the antelope…
Benoit proposes: (1) Nothing like a good night (2) Sorry for the back (3) First task: bathing!
The subjects once again chooses their preferred storyline:
Vincent chooses Paul's text.
Paul keeps his text.
Francesca choses Paul's text.
Benoit keeps his text.
Now they propose a text for the fourth panel, the "punchline" of the story.
Vincent proposes: *(1) The Smooth Slope… (2)…the Easter Egg… (3) The One Beer for me…* (4)… Yoga for cats
Benoit proposes: *(1) Nothing like a good night (2) Sorry for the back (3) First task: bathing!* (4)… And also for today, I am done working!
In the final step, the subjects have to vote for the best story: the consensus is complete, as everybody votes for Paul's ending.
Figure 2 illustrates the whole process visually:

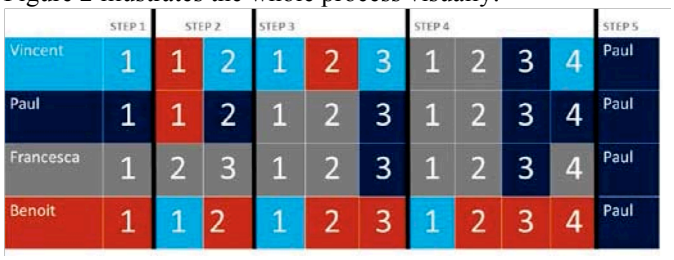


Figure 2: the process of creation and choice, on 5 steps. Each colour corresponds to a player.

2 Results

We recorded, at each step, whether the subjects have pursued their own story or switched to the story of someone else. In addition to this "switching behavior", we also recorded the number of votes each participants received at each step. This measurement was used as a quality level of each text, and the mean number of vote received as a performance level for each participant.

2.1 Description of the population

Among the 953 individuals who registered to the experiment's website, 756 subjects did not complete a single session and were excluded from analysis, leading to a sample of 197 subjects who completed at least one session. The high difference between registered users and actual subjects may be due to the fact that seldom four potential subjects sharing the same language and the same strips to be completed would be online at the same time.
Figure 3 illustrates the number of participants according to the number of experiences completed, where it can be observed that the majority of subjects concluded only one session. This may be due to the length of the waiting time between sessions to find other participants or to the perceived difficulty of the task.
Among this population, the mean age is 33.7 years, with a standard deviation of 11.9 years. The youngest subject is 11 years old and the oldest one is 73.
Gender is equally distributed among the subjects, with a 53% of males and 47% females.
The experiment was available in five different languages (English, French, Italian, Dutch and Spanish) and the distributions of the native languages and fluent languages are illustrated by Figure 4.

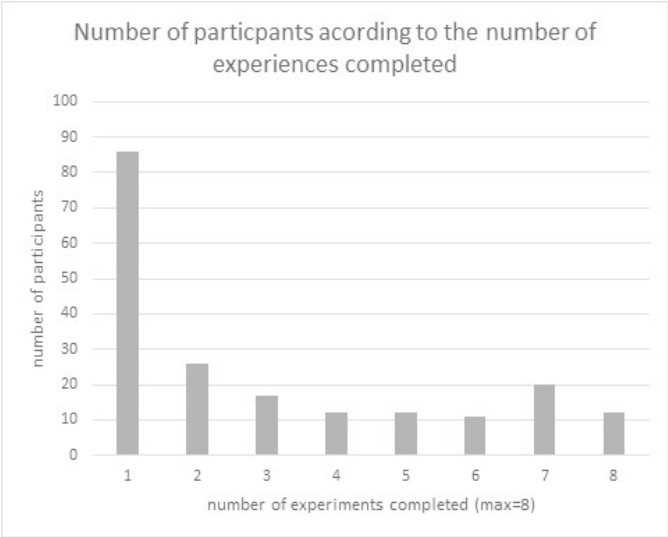
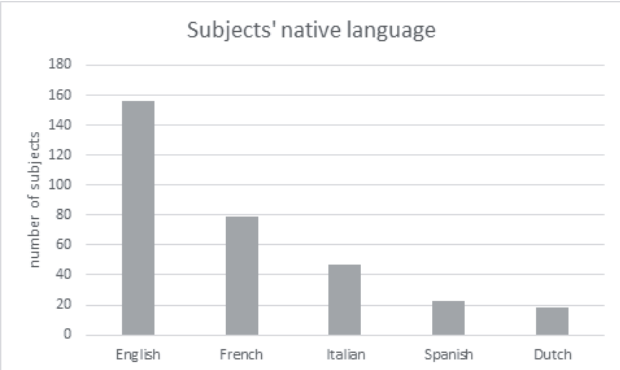


Figure 3: attendance to sessions



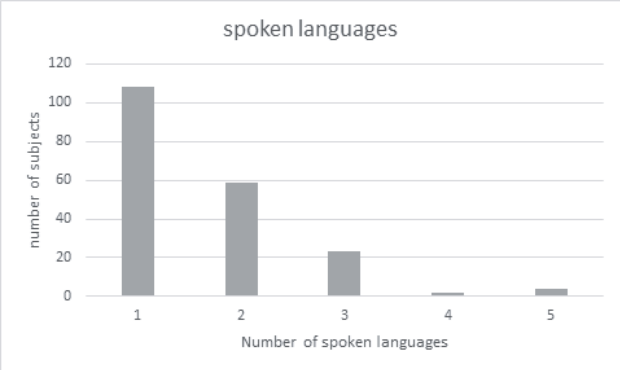


Figure 4: Distribution of native and spoken languages among the population

2.2 Emerging profiles: switchers and pursuers

Table 5 illustrates the subjects' behaviour during each step implying a decision (switching in favour of another person's text or pursuing their text), thus identifying the emerging "profiles types". It occurs that "extreme" profiles are the most common ones: the most frequent behaviour corresponds to the profile type 1, or the "perfect pursuer",

Profile type	Switching behaviour	Number of subjects (mean profile)	Number of subjects (exact profile)
1	0000	44	20
2	1000	25	14
3	0100	10	2
4	0010	7	1
5	0001	5	1
6	1100	6	2
7	1010	5	3
8	1001	4	3
9	0110	4	3
10	0101	2	1
11	0011	1	0
12	1110	17	6
13	1101	9	3
14	1011	5	2
15	0111	4	3
16	1111	48	17
	Total	196	81

Figure 5: profile types and their distribution among the population. In the second column, 0=pursue and 1= switch

who never switches for another subjects' content. It is followed by the "perfect switcher" profile, who keeps abandoning her/his own production in favour of the production on someone else. In third position (see Figure 6) we find those subjects who switch only once, during the first step, and then pursue their production. Because subjects can participates to several experiments (up to eight) their profiles are not necessary systematics. For example, a subject could choose not to switch during the first experiments, but could choose to switch during the others.

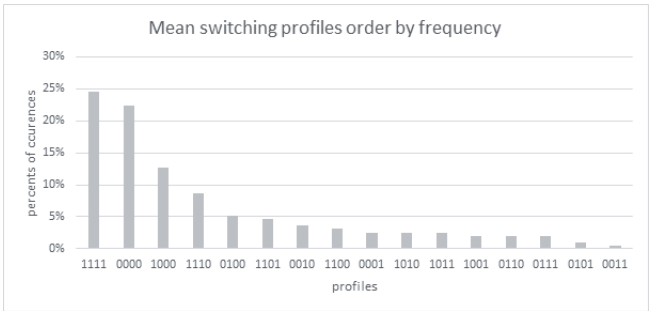


Figure 6: The most frequent profiles

We observe (Figure 7) that the three-step profile's distribution is very similar to the distribution taking into

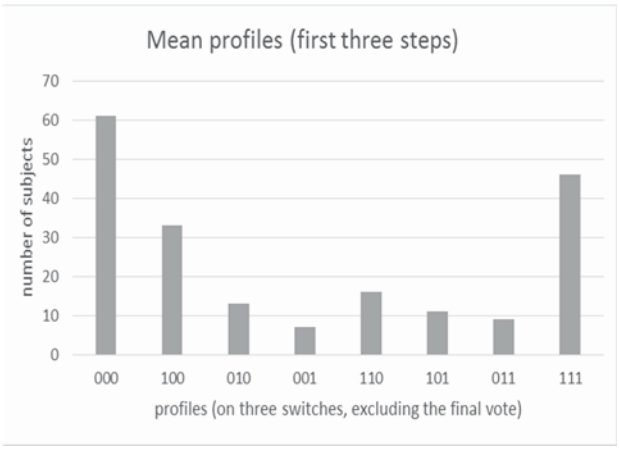


Figure 7: Profiles distribution concerning the first three steps

account the four steps: the most frequent profiles are the perfect switchers and the perfect pursuers, followed by the ones who switch only during the first step. Theses results suggest that the profile distribution is consistent, reliable and is not an artefact due to a windowing effect.

2.3 Switching stability

By observing the total switching rates, we can determine that the switch and pursue rates are coherent and constant throughout the different experiments. This means that the number of sessions to whom each subject participated does not impact the switching rates, as illustrated in Figure 8.

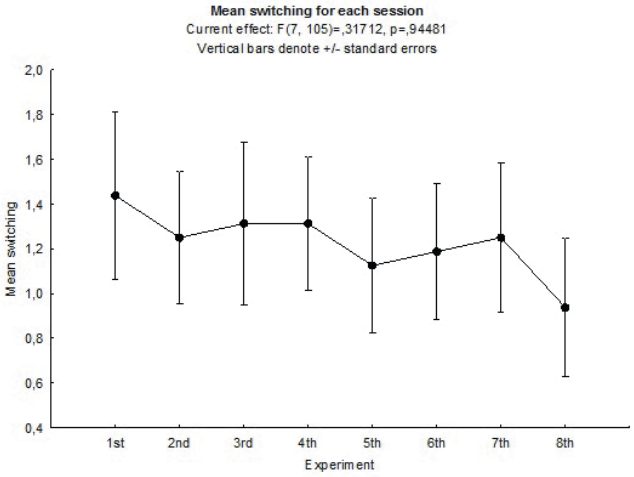


Figure 8: the switching mean remains stable throughout the eight experiments.

On the other hand, we can see that the number of switches significantly decreases inside the sessions: subjects switch less and less at each evaluation step (F[3,522] = 11,084, p<.001), as illustrated in Figure 9:

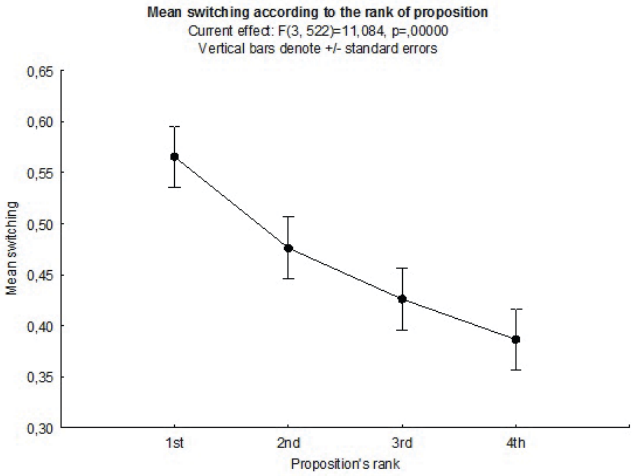


Figure 9: subjects' tendency to switch significantly decreases at each step

2.4 Votes

As mentioned above, subjects can either switch or continue their own text, knowing that the very text they produced can be chosen by other participants, whether or not the author is a switcher. We refer to the event of a text being chosen by someone else as a "vote".

Figure 10 illustrates the mean of votes received for each step by the "pursuers" (the 50% subjects who pursue the most in light grey) and by the "switchers" (the 50% subjects who switch the most in dark grey).

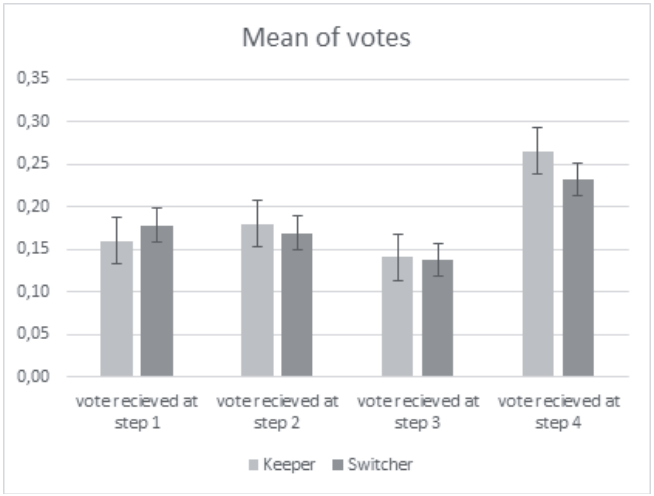


Figure 10: votes received by pursuers and votes received by switchers

Votes received by pursuers and votes received by switchers are not significantly different (F[8,196]=1,390, p>.10). This results stays true even on more contrasted groups such as the "perfect switchers" and the "perfect pursuers".

The number of votes received by a text which has been "abandoned" by its author (mean =0.87) is not significantly different from the number of votes received by a text which has been kept by its author (mean = 0,76) (t[164] = 0.790 ; p = .43). This result stays stable for each step.

2.5 Returners

We can identify another common behaviour, the one of the "returners". The returners are those switchers who abandoned their first and/or second panel's position productions and recover them if another subject "adopts" them. Among the 487 stories where the first panel's text was abandoned by its author, only 130 (27%) were abandoned also by everybody else . On the 357 stories remaining, (where, therefore, a "return" was possible), we observed 109 (31%) returns, 165 (46%) continuations where the original author did not return and decided to pursue his/her production and 83 (23%) switches where the original author did not return and decided to switch again for the story of another subject.

Among all texts the repartition of returns is as follow: 14% have been abandoned at step 1, voted by one or more

subjects and then recovered at stage 3 by their original authors (returns); 5% have been abandoned at step 1, voted by one or more subjects and then recovered at stage 4 by their original authors; 15% have been abandoned at step 2, voted by one or more subjects and then recovered at stage 4 by their original authors. This results show that the returns are more likely to occur when the initial switch (abandon) and the return are separated by only one panel.

2.6 Gender and language effect

Results show no effects between male and female participants, whether it is for the switching rates ($t[191] = 0.40$; $p = .31$) or the number of votes received ($t[191] = 1.17$; $p = .22$).

Because the task was available in five different languages, we also tested the effect of language and found no significant differences between the number of spoken languages and the switching rates ($F[4,191] = 1.03$; $p = .39$) or the number of vote received ($F[4,191] = 1.37$; $p = .24$). No differences were observed also between the participants' mother tongue and the switching rates ($F[4,187] = 1.38$; $p = .24$) or the mean number of vote received ($F[4,187] = 0.14$; $p = .96$).

3 Discussion

3.1 Self-evaluation bias or engagement?

In accordance to previous studies [Kaufman and Evans 2010] [Priest 2006], our findings challenge the validity of self-assessments in creativity. This result stems from the decrease of the switching rate for the later steps, which is significant even if the subjects were instructed and motivated to choose the best story independently from their contribution to it: this means that, at the end, a majority of subjects judged their own story as the best one. The motivations for this observed behaviour could be a selfenhancing bias and/or a progressive engagement in one own' work.

The self-enhancing bias, or self-serving bias, is the tendency to perceive oneself more positively than a normative criterion would predict [Krueger 1998]. This could be the explication of the significant tendency to chose one's own creation as the best one in the final steps. Regarding the outcomes of the first steps, apparently in contradiction with the self-enhancing bias theaory, they are consistent with the exploratory behavior usually observed in creativity tasks [Finke, Ward & Smith, 1992]. Aself-serving bias specifical to the creative process is a result that has never been highlighted before, to our knowledge.

Another motivation for the significant decrease of the switching rate in the later steps may also be the increasing effort that subjects have applied in the creative process. The design of the experiment itself, proposing a sequential creative activity, makes possible to highlight this occurrence. Subjects are indeed more open to switch at the beginning of the process, but as they put effort and invest their time in the task, they become more attached to their production, as a commitment effect [Beauvois, Joule & Brunetti, 1993]. This result also implies that it is easier to change the direction of a creative work at its early stages rather than towards the end. To our knowledge, this result has never been captured by a scientific experiment; nevertheless, [MacKinnon 1978] highlighted that experienced architects are more likely to abandon their ideas than beginners. This could mean that, independently from its motivation, being aware of the bias "against change" is a skill of the creative professional.

3.2 Profiles and quality of the outputs

We could not observe any significant impact of the profile type on the quality of the productions, evaluated by the number of votes received ($F[7,188] = 1.03$, NS). We thus tried to analyse patterns in the ten strips that received the most consensus. Four of them received a unanimous consensus from all four participants, and six from three out of four. We determined that three of them were composed by only one author who never switched (and it should be noted that two of these stories were made by the same author, indicating a very creative participant), six were composed by two different authors and one by three different authors. This result [FG6] suggests that consensus can be more easily attained when stories are created through collaboration (using or giving ideas from/to others). Interestingly, we can see that on the seven experiments where there was a consensus of three participants, the nonconsensual response by the fourth participant was on six times out of seven for his or her own production. This reinforce the result that the tendency of selecting one's own story during the final step is so strong that participants prefer to do so even when there is a worthwhile story. We can also observe that the profile distribution of the subjects who has drawn the most votes from their coparticipants is the same as the whole sample. This confirms that the switching or pursuing profile is not linked to the quality of the productions.

Another interesting result concerning the profiles is that the most frequent profiles are "extremes", that is participants who never switch or always switch. One explanation would be in term of personality, mainly the openness dimension with its tendency to explore other ideas, to try something new. Another one would be in term of self-esteem, where subjects with a low self-esteem would consistently judge, and hence chose, other stories better than their own production, whereas subjects with high self-esteem will do the opposite. These suggestions are purely speculative, and it would be interesting to replicate this study with a personality questionnaire and a self-esteem evaluation to assess them.

3.3 Language effect

Interestingly, the results concerning language seems to contradict the classical advantage of multilinguals on creativity tasks, where they usually outperform the monolinguals [e.g., Karapetsas and Andreou, 1999;

Kharkhurin, 2008]. However, the absence of effect could depend from the self-evaluation, since productions were assessed by the groups themselves through the number of votes each text received.

3.4 Effect of implicit feedback and "returning" behavior

In the context of the Comic Strip Game, the subjects have to select which story they will continue. This implicitly indicates to the subjects that they will receive feedback on their production, not only at the end of the strip as mentioned in the instructions, but also at the end of each panel, in a within-group evaluation which should deflect, as explained above, most of the social loafing effect [Szymanski & Harkins, 1992]. Indeed, steps 3 and 4 provide the participants with implicit feedback on their previous productions because they can see if their text has been chosen or not by other subjects. This implicit feedback can be particularly interesting when a subject has abandoned his or her first production(s) and then realises at step 3 or 4 that someone else has selected his/her "abandoned" text. This can lead to a cognitive dissonance for the author of the first text because he or she judged it not good enough, or with less potential than other texts, but others saw instead a good idea or an interesting potential [Festinger, 1962].

We observed that authors who have abandoned their text at the first panel (i.e. they have switch at the first step) and could return to continue their story later, in a large majority they did not (69%). In other words, once an idea is discarded, it is for good and reconsidering it is less likely to happen. This result is consistent with the commitment effect described earlier which suppose a consistency from previous choices and a difficulty to change opinion or judgement, particularly when the choice was voluntary and not constraint. Moreover, we can see that this effect is more important when the delay between the first switch and the return is longer, because when the delay is one panel long (between first and third step or between second and fourth step), we have a constant return rate of 15%, while only 5% when it is two panel long (between first and fourth step).

These results suggest that the feedback provided by the votes of the other participants are not enough to compensate the "anti-change" bias.

3.5 Conclusion and future work

The data analysis of the Comic Strip Game has given us an insight on the creative process. Our results highlight that the potential impact of implicit feedback from other participants and objectivity in self-evaluation, even if encouraged, are lessened by a bias "against change". Such a bias probably stems from a combination of selfenhancing bias and of a commitment effect. We could also highlight consistent and stable strategies for content creation which, interestingly, are not related to gender, age and spoken languages.

Future work may focus on the "against-change bias", for example to test whether it resist to within group social pressure, and to explicit, external or internal feedback. It may be also interesting to design a protocol investigating the motivations of the bias, in order to distinguish between the self-serving illusion and the commitment effect.

Acknowledgments

This research is conducted within the Flow Machines project, which received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 291156.

We thank Johann Girod for implementing the online system which allowed us to run the experiment and Naomi Ziv for the insightful discussions about future work.

Note: All completed strips are downloadable at this address: http://www.fiammettaghedini.com/evaluation-comics

References

[Amabile 1979] Amabile, T. M., Effects of external evaluation on artistic creativity, Journal of Personality and Social Psychology, Vol 37(2), Feb 1979, 221-233

[Bartis 1988] Bartis, S., Szymanski, K., & Harkins, S. G. (1988). Evaluation and performance: A twoedged knife. Pers Soc Psychol Bull June 1988 vol. 14 no. 2 242-251

[Beauvois 1993] Beauvois, J.-L., Joule, R.-V., & Brunetti, F. (1993). Cognitive rationalization and act rationalization in an escalation of commitment. Basic and Applied Social Psychology, 14(1), 1–17.

[Festinger 1962] Festinger, L. (1962). A theory of cognitive dissonance (Vol. 2). Stanford university press.

[Finke 1992], Finke, R. A., Ward, T. B., & Smith, S. M. (1992). Creative cognition: Theory, research, and applications.

[Freeman 2014] Freeman,O., Tangney, D., O'Rourke, B.K., (2014): Performing Collaborative Creativity: Learning from Diverse Experts Interacting in Ireland's Science Gallery, 30th European Group for Organizational Studies Colloqium, Rotterdam

[Karapetsas 1999] Karapetsas, A., & Andreou, G. (1999). Cognitive development of fluent and nonfluent bilingual speakers assessed with tachistoscopic techniques. Psychological Reports, 84, 697–700.

[Kaufman and Evans 2010] Kaufman, J. and Evans. M., The American Idol Effect: Are Students Good Judges of Their Creativity across Domains? Empirical Studies of the Arts January 2010) vol. 28 no. 1 3-17

[Kharkhurin 2008] Kharkhurin A. V. (2008). The effect of linguistic proficiency, age of second language acquisition, and length of exposure to a new cultural environment on bilinguals' divergent thinking. Bilingualism: Language and Cognition, 11, 225–243.

[Krueger 1998] Krueger, B, Enhancement bias in Descriptions of Self and Others, PSPB, Vol.24, No5, May 1998, pp. 505-516

[Lubart 2013] Lubart, T., Mouchiroud, C., Tordjman, S. & Zenasni, F. (2003). Psychologie de la créativité [Psychology of creativity), Paris, Armand Colin

[MacKinnon 1978] MacKinnon, D. W. (1978). In search of human effectiveness: Identifying and developing creativity. Buffalo, NY: Creative Education Foundation.

[Priest 2006] Priest, T., Self-evaluation, creativity, and musical achievement, Psychology of Music, January 2006, vol. 34 no. 1 47-61

[Silvia and Phillips 2004] Silvia P. and Philips A. 2004, Self-Awareness, Self-Evaluation, and Creativity, Pers Soc Psychol Bull August 2004 vol. 30 no. 8 1009-1017

[Simms 2014] Simms, A., (2014) Social Loafing: A Review of the Literature, Journal of Management Policy and Practice vol. 15(1)

[Szymanski & Harkins 1992] Szymanski, K., & Harkins, S. G. (1992). Self-evaluation and creativity. Personality and Social Psychology Bulletin, 18(3), 259-265.

Annex 1

Improving music composition through peer feedback: experiment and preliminary results

Daniel Martín and **Benjamin Frantz** and **François Pachet**
Sony CSL Paris
{daniel.martin,pachet}@csl.sony.fr

Abstract

To which extent peer feedback can affect the quality of a music composition? How does musical experience influence the quality of a feedback during the song composition process? To answer these questions we designed and conducted an experiment in which participants compose short songs using an online lead sheet editor, are given the possibility to feedback on other participant's songs and can either accept or reject feedback on their compositions. This experiments aim at collecting quantitative data relating the intrinsic quality of songs (estimated by peer evaluation) with the nature of feedback. Preliminary results show that peer feedback can indeed improve both the quality of a song composition and the composer' satisfaction about it. Also, composers tend to prefer compositions from other musicians with similar musical experience level.

1 Introduction

Peer feedback has become an ubiquitous feature of online education systems. Peer feedback consists in letting students or participants in a class revise, assess and more generally comment on the work of other students. This model is opposed to the traditional one in which students' works are evaluated only by a teacher. Peer feedback is acknowledged to bring many benefits [Rollinson, 2005] such as saving teachers' time as well as other pedagogical positive effects [Sadler and Good, 2006]. With the increase of online learning communities and MOOCS [September, 2013], peer feedback is becoming more and more popular.

Peer feedback is not only useful in pedagogical contexts, it can be also used in creative tasks. In music composition, collaborative composition has been addressed in several studies [Donin, forthcoming 2016]. There are online creative communities in which music is composed collaboratively by several users [Settles and Dow, 2013].

In those creative contexts, the following questions are legitimate: to which extent peer-feedback can affect the quality of a musical composition? What is the influence of the musical experience of the composers involved in this process? To address these questions we have designed a music composition experiment based on anonymous one-way feedback with no dialogue. Such a scenario differs from typical collaborative composition contexts in which composers work together hand by hand in a composition. The experiment is not aimed at being realistic or to propose a new tool for collaboration composition, but specifically to collect quantitative data regarding the relation between feedback, skills and song quality.

We focus on the role of peer feedback in music composition, specifically in *lead sheet* composition. A lead sheet is a representation of a simple song consisting of a melody and a corresponding chord grid. We propose an experiment in which peer feedback consists in suggestions of changes of certain parts of the lead sheet: specific notes or groups of notes or chords. These musical suggestions can be accompanied by a text explanation. Once a feedback is posted by a participant, it can be reviewed by the composer, who then decides to either accept it (and modify the lead sheet accordingly) or discard it.

Additionally to the sheer effect of feedbacks, we also examine the characteristics of the composer, commentator or judge of the participants. Indeed, having an extended experience in music composition might be seen as a prerequisite to write a nice song or to give useful suggestions. However, previous research showed that expertise might not be as critical as we could expect [Frese *et al.*, 1999].

2 Description of the experiment

Participants are instructed to write a short composition using an on-line lead sheet editor [Martín *et al.*, 2015]. Then they are asked to give feedback to another participant's composition, and finally they are asked to improve their own original composition using feedback posted on their composition. Participants are divided randomly in two groups: participants in the control group (G1) do not receive any feedback, and try to improve the song by themselves, whereas participants from the experimental group (G2) may use the feedback received to improve their own song. The existence of these two groups is ignored by the users so that the results are not biased.

As we are trying to assess the impact of feedback on the quality of a music composition, we need to estimate the *quality* of all compositions as well as their various variations during the experiment. To do so we use social consensus to de-

termine the quality of a song: participants listen and are given the possibility to "like" other participants' compositions. The quality of a song is then simply determined by the number of likes obtained for that song. In the next section we describe in detail each phase of the experiment:

2.1 Questionnaire

Participants start the experiment by answering 15 questions about to their experience in music, and more specifically in music composition. For example, they are asked how many years they have studied music theory, how many years they have been playing in a band, which style of music they like more, how often do they compose... etc.

2.2 Original composition

Participants then write a short composition using the online lead sheet editor. A lead sheet is a particular type of music score widely used in jazz, bossa-nova and song-writing, consisting on a monophonic melody and a chord grid. All compositions have a fixed length of 8 bars; participants are not able to add or delete bars, but they can choose the tempo and the time signature of the song. Participants fill the 8 bars with a melody and chord labels (e.g. Dmaj7, Em7...etc.). Figure 1 shows a screen-shot of the lead sheet editor.

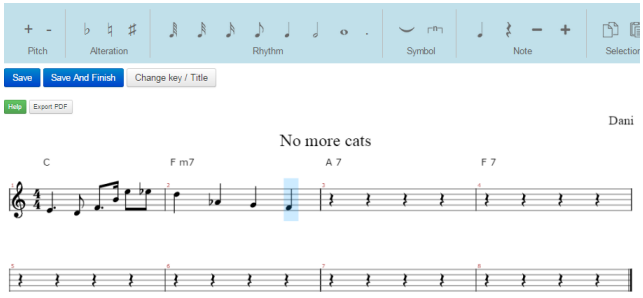


Figure 1: Screenshot of a composition being entered with the lead sheet editor.

Participants can listen to their composition with a basic MIDI player. When they are done they click on "Save and Finish". Next, they answer a questionnaire about their confidence in the quality, complexity and satisfaction on their composition.

2.3 Feedback Posting

Once they have finished their composition they are asked to give feedback to another participant by suggesting improvements in another participants' composition. Each suggestion can be at the most, two bars long. Participants can make as many suggestions as they want as long as they do not overlap. So, each participant can make a maximum of 8 suggestions (one per bar). To make a suggestion, participants must choose the bar(s) to modify, then they can change the notes and the chord symbols. Optionally, they can also leave a text comment explaining their changes. Figure 2 shows a composition in which a participant is entering suggestions with an explanation. When they are finished, they answer a short questionnaire about their confidence on the suggestions they just made as well as their opinion on the original song they modified.

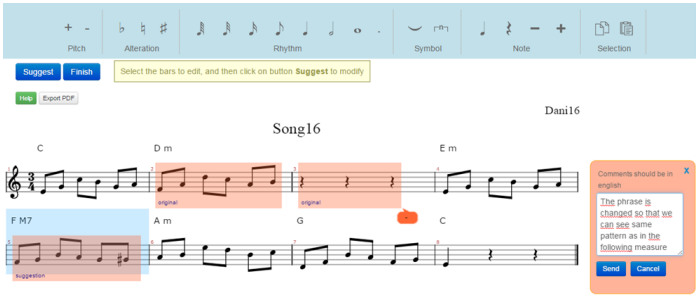


Figure 2: Screenshot showing a participant entering an explanation of the suggestion.

2.4 Improvement: Final composition

Next, participants are asked to reconsider their own composition and are asked to try to improve it. Participants from G1 (control group) are told that they unfortunately did not receive suggestions and are encouraged to try to improve their own composition by themselves. Participants from G2 see the suggestions they received from two other participants. They can listen to all the suggestions. If they like a suggestion they can *accept* it, so that it is kept and the song is automatically updated accordingly. In addition to integrating suggestions, they can modify freely their composition. Once they are finished, they answer a questionnaire about their confidence on their own improvement and on their opinion on the suggestions received.

2.5 Evaluation phase

The last step of the experiment is to evaluate pairs of compositions from other participants. Each pair of songs consist on the original song and the improved song. Participants are asked to evaluate each song by place it in a vertical display with a legend from 0 ("I don't like it") to 100 ("I like it very much"). Participants do not know which is the original and the improved song when they are evaluating. One of the versions is presented as *song A* and the other as *song B* and this assignment is performed randomly. Participants have to evaluate at least 5 pairs of songs in order to finish the experiment.

3 Results

In this section we describe in detail the results obtained from each phase of the experiment.

3.1 Population

The experiment was conducted between February and July 2015. 66 participants completed the experiment (68% men and 32% women). Mean age was 29.2 years, ranging from 19 to 61. Musical experience was measured through a questionnaire with 7 items. The scale has a satisfactory sensibility with an observed range from 7 to 41 (out of 0 to 42) and we observed a mean of 28.7 with a Standard Deviation (SD)

of 8.9. The intern consistency is satisfactory (Cronbach's alpha=.82).

Composition experience was measured through a questionnaire with 5 items. The results show an overall low level of experience concerning composition in our sample with a mean 6.9 (SD=6.1) on a scale ranging from 0 to 30). The intern consistency is satisfactory (Cronbach's alpha=.85).

3.2 Composition effects

Each participant was randomly assigned to either the control group (G1) or the experimental group (G2). No significant differences were observed between the two groups in relation to age, gender, musical experience or composition experience.

Composition evaluations

During the evaluation step, we checked if participants had listened to the songs before evaluating them. On the 1195 evaluations made, 219 were made without listening to the song. We removed those evaluations.

The songs were evaluated by an average of 8.8 different judges. The mean score of the evaluations made during the evaluation phase is 53.25 (SD = 13.26) on a scale ranging from 0 to 100. However, judges might be more or less strict, and some songs might have been evaluated by a particularly strict or generous participant. To take into account the severity of the judges, we have standardized the evaluations to get z-scores where the mean and standard deviation used are based on all the evaluations made by a given participant. As a result, the mean of the standard scores is approximately equal to zero, and a standard deviation of approximately .50. It should be noted that this final score correlates strongly with the raw score (r=.84). This result indicates that we had enough evaluations for each songs to avoid any severity bias.

Original Composition

The questionnaire that participants were asked to complete after finishing the original composition included self-estimation questions about the quality, complexity and satisfaction for their composition on scales ranging from very bad/simple/unsatisfied (0) to very good/complex/satisfied (6). We also asked them to evaluate the time they spent to make their composition and if they used an instrument to help them to compose (and which instrument if they did).

Results show a mean quality of 2.8 (SD=1.5), a mean complexity of 1.9 (SD=1.6) and a mean satisfaction of 3.2 (SD=1.6). Only the complexity is significantly different to the center of the scales which is 3 (T(65)=-5.27 ; $p<.0001$). This means that the participants tend to judge their work as rather simple (low complexity). We also observed positive and significant correlations between these three measures, ranging from r=.41 to r=.80.

During the suggestion step, we asked the participants to also rate the quality and complexity of the songs they had to comment. Each composition from the experimental group (G2) was commented by two different participants. In the end we obtained the score from the author and two other scores from two different commentators. Interestingly, there was no correlation between the scores from the original composer and the ones from the commentators (r<.10), but the two commentators did agree together on the quality (r=.80) and on the complexity (r=.70).

Moreover, from the judgments done during the evaluation phase (in which participants evaluate pairs of songs from other participants), the measurement of the quality of each original song (standardized to z-scores) allows us to estimate the composition skills level of its author. Surprisingly, we observed that the quality of the original song is only marginally related to the composition experience (r=.18, p=.15) or to the musical experience (r=.19, p=.12).

We also asked the participants whether they used an instrument to help them in their composition. Results show a marginally significant effect in favor of the use of an instrument on the mean quality score ($T(64)$=-0.87, p=.38).

The mean duration of the composition time of the song as evaluated by the participants is 30 minutes (SD=32 min) ranging from 1 minute to 240 minutes. This evaluation is largely underestimated by the participants because the real duration calculated from the time spent on the composition software is significantly longer (m=67 min; $T(65)$=4.20, $p<.001$). The correlation between these two durations is not very high, but significant (r=.46, $p<.001$) indicating that the error of duration estimation is not exactly the same for everyone. Interestingly, we observed that the quality of the original songs (from the evaluation phase) is not linked with the time spent to compose, whether it is subjective (r=.04) or objective (r=.03). This result suggests that in a situation where there is no time constraint, the amount of time devoted to compose has no effect on its quality.

Finally, there is a difference in the consensual quality of the original song, obtained from the evaluation of several participants (0.07 in G1 vs. -0.15 in G2). This could be due to differences in the group of judges evaluating each song.

Suggestions

In the questionnaire filled after making the suggestions, participants were asked how much do they think the song they are revising will be improved due to their modifications (on a 7 points Likert scale ranging from 0 "very little", to 6 "very much").

The participants from G2, the experimental group (N=30), received two suggestions for their final composition. Once they finished, we asked them if the suggestions received were interesting (on a 7 points Likert scale ranging from 0 "very little", to 6 "very much"). Additionally, we recorded the number of suggestions they received and the number of texts comments received.

We ran a series of correlations between these measures and the improvement effect (the difference between the original song and the final song on the quality judgment score). None were significant, suggesting that neither the number of suggestions received nor the number of explanations for that suggestions have an impact on the improvement of a song.

Final composition

Overall, we can see that the control group, G1, does not improve significantly between the original song (m=.07) and the final song (m=.12) (improvement effect = .05, $T(35)$=0.94, p=.35). However, we do see a significant improvement for

the experimental group, G2, between the original song (m=-.15) and the final song (m=.08) (improvement effect = .23, $T(29)=2.47$, $p=.02$). See Figure 3.

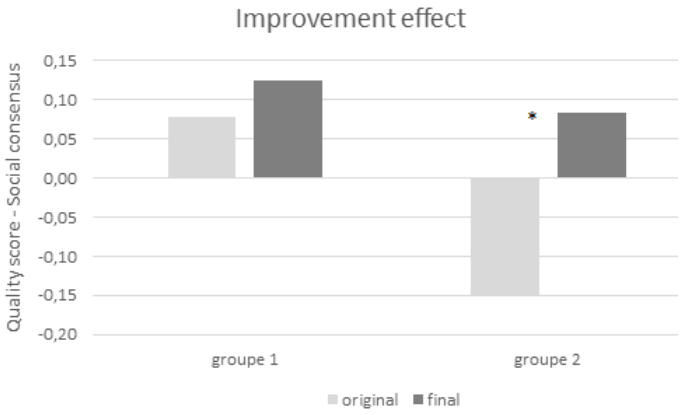


Figure 3: Difference between the original song and the final song on on the quality judgment score for the group without feedbacks (G1) and the group with feedbacks (G2).

We also examined the subjective evaluation of the participants concerning the improvement of their song. We constructed two composite scores. One from the self-evaluation scales of the original song (quality, complexity and satisfaction), one from the self-evaluation scales of the final song (quality, complexity and satisfaction). The intern consistency of those composite scores are satisfactory (the two Cronbach' alphas are above .81). We then conducted a mixed *between participants* (control and experimental groups) x *within participants* (original and final song) analysis of variance. We observed a significant interaction between groups and songs ($F(1,64) = 7.07$, $p=.01$). To explore this interaction, we used a post-hoc analysis with Tuckey HSD tests. Results show that participants who received suggestions had a significant improvement between the original and final song ($p<.001$) while the control group had no improvement ($p=.49$) See Figure 4.

When evaluating songs, users did not know which song was the original and which one was the final, as the order of the songs was determined randomly. This was a design decision to avoid the fact that participants could tend to rate better the final song, as it is supposed to be improved. Aditionally we wanted to ensure that songs were not better rated just because they had more modifications. To check this point, we used and melodic similarity algorithm [Urbano *et al.*, 2011] to estimate the similarity between each original and final songs. The correlation between the percent of similarity and the improvement effect based both on the composer's subjective opinion and on the scores from the judges are low (r=-.36, p=.003 and r=-.19, p=.13), which suggests that the improvement is not linked to the dissimilarity between the two versions.

Lead sheet editor

The software used was developed specifically for the experiment and we asked participant whether it was frustrating (0) or helpful (6) to compose with it. Results show a mean of 3.13 after the first composition and 3.41 after the final composition (the difference is not significant) which means that even if the online editor was not specially helpful, it did not hinder the composition process.

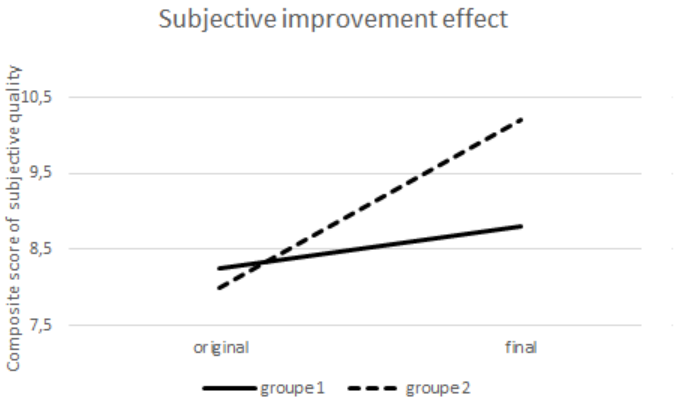


Figure 4: Self-esteemed quality of the original and final songs for the group without feedbacks (G1) and the group with feedbacks (G2).

Experience effect on evaluations

To find out whether musical experience has an impact on the way participants judge song from other participants. We divided our sample of participants in two groups according to their experience as musician (based on the median). We also divided our sample of songs according to the experience as musician of their author. We then ran a two-way ANOVA to explore the effect of the experience of the judges according to the experience of the compositor. Results show a crossed interaction between these two variables ($F(1,61)=7.63$, $p=.007$) as illustrated in figure 5. These results indicate that experienced judges give high scores to songs from experienced authors and low scores to songs from non-experienced authors. It is exactly the opposite for the non-experienced judges. This means that participants tend to prefer compositions from other participants with similar experience. This could explain the difference in the evaluation of the original songs in G1 and G2. The groups of judges evaluating each song could have different level of expertise.

4 Conclusion

The aim of this experiment was primarily to examine quantitatively the impact of peer feedback in music composition and secondly to assess how important is the experience of the participants as musicians or composers in the whole process. Before any improvement or suggestions, participants had to write their first song. Interestingly, results show that participants' previous experience in composition did not impact the quality of their song. The same pattern was also found for the participants' previous experience as a musician. These two results suggest that the quality of a song (based on social consensus) does not really tap in musicality but in something

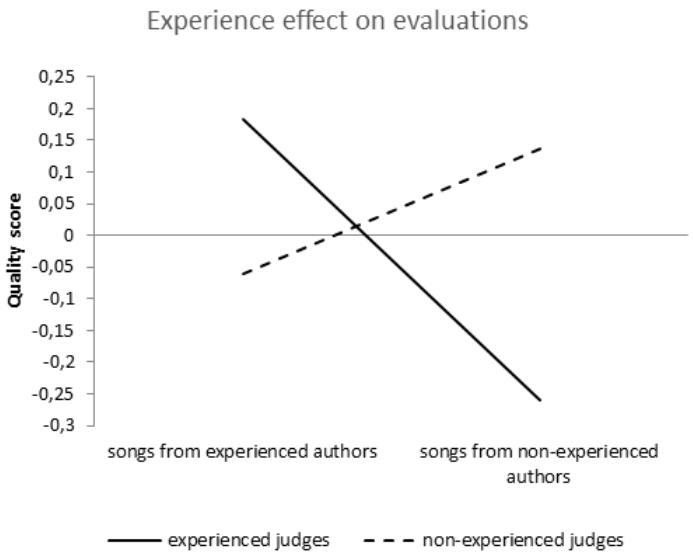


Figure 5: Interaction between the experience of the author and the experience of the judges on the quality score.

else, presumably creativity. As before, creativity might play an important role [Frese *et al.*, 1999].

Results show that composers who received feedback (G2) clearly evaluated better the improved song than the original, meaning that they were satisfied with the improvement they made. Further, the evaluation based on social consensus had a longer improvement also for G2. Hence, participants who received feedbacks not only felt that they had composed a better song after the improvement step, but they actually did. This basic finding suggests that improvements in music may be achieved even without real collaboration with dialogues and active interactions, but by simple suggestions on a single occasion.

Since there is a difference on the evaluation of the original songs between G1 and G2, we wanted to verify whether experience can make a difference when evaluating songs and we found out that participants tend to like more songs that are composed by other participants with similar musical experience.

Future work may be done by going deeper in determining the influence of the participants' experience. For example, by checking when are songs more improved, taking into account the experience of composers, commentators and judges. Further, we could assess more precisely which suggestions were actually used (or accepted) by the original composer to obtain a ranking of commentators whose suggestions are most accepted, as a measure of how good commentators they are. We could check also if suggestions from experienced commentators are more likely to be used from inexperienced composers, or whether experienced composers usually accept suggestions of other composers, and how does this affects the improvement of the song.

Acknowledgments

This work is supported by the Praise project (EU FP7 number 388770), a collaborative project funded by the European Commission under programme FP7-ICT-2011-8.

References

[Donin, forthcoming 2016] Nicolas Donin. Domesticating gesture: the collaborative creative process of florence baschet's streicherkreis for 'augmented' string quartet (2006-2008). *Eric Clarke & Mark Doffman (eds.), Creativity, Improvisation and Collaboration: Perspectives on the Performance of Contemporary Music, New York: Oxford University Press*, forthcoming 2016.

[Frese *et al.*, 1999] Michael Frese, Eric Teng, and Cees JD Wijnen. Helping to improve suggestion systems: Predictors of making suggestions in companies. *Journal of Organizational Behavior*, 20(7):1139–1155, 1999.

[Martín *et al.*, 2015] Daniel Martín, Timotée Neullas, and François Pachet. Leadsheetjs: A javascript library for online lead sheet editing. In *First International Conference on Technologies for Music Notation and Representation (TENOR)*, Paris, France, 2015.

[Rollinson, 2005] Paul Rollinson. Using peer feedback in the esl writing class. *ELT journal*, 59(1):23–30, 2005.

[Sadler and Good, 2006] Philip M Sadler and Eddie Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.

[September, 2013] On September. Behind the scenes with moocs: Berklee college of musics experience developing, running, and evaluating. *CONTINUING HIGHER EDUCATION REVIEW*, 77:137, 2013.

[Settles and Dow, 2013] Burr Settles and Steven Dow. Let's get together: the formation and success of online creative collaborations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2009–2018. ACM, 2013.

[Urbano *et al.*, 2011] Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. Melodic similarity through shape similarity. In *Exploring music contents*, pages 338–355. Springer, 2011.

[Van den Berg *et al.*, 2006] Ineke Van den Berg, Wilfried Admiraal, and Albert Pilot. Designing student peer assessment in higher education: Analysis of written and oral peer feedback. *Teaching in Higher Education*, 11(2):135–147, 2006.

When Reflective Feedback Triggers Goal Revision: a Computational Model for Literary Creativity*

Pablo Gervás
Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
28040 Madrid, Spain
pgervas@sip.ucm.es

Carlos León
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain
cleon@fdi.ucm.es

Abstract

Existing models of the writing task from a cognitive viewpoint agree on the importance of draft revision in the overall process. This is generally assumed to focus on reviewing intermediate drafts in search for feedback on how to modify them to match the driving constraints. However, in literary creativity it is often the case that the feedback leads not to a revision of the current draft but to a redefinition of the constraints that are driving the process. This phenomenon is explicitly described in Sharples' model of writing as a creative task. Yet existing computational models of literary creativity do not contemplate it. The present paper describes a computational model of the creative processes in literary creativity that contemplates the explicit representation of the constraints driving the process, and allows for the feedback from the validation to modify not just the ongoing draft but also the constraints that it is expected to satisfy. This allows the model to represent cases of serendipitous discovery of interesting features.

1 Introduction

Creative processes as carried out by humans are known to involve a significant amount of trial and error. Writers, musicians, painters, poets... rely on a succession of drafts that get polished over many iterations, each one involving feedback from the previous version, and resulting from a process of revision or regeneration of it. Yet computational models developed in AI over the years to emulate these same processes very rarely capture this type of dynamic operation. Sometimes they do, in a limited fashion, when an AI program includes an evaluation function that defines the desired form for outcomes, and this is run over the results of a generative process that produces candidate artefacts of the desired type. However, the dynamics of the creative process in humans are known to be more complex than this, possibly differing significantly across different domains.

*This paper has been partially supported by the project WHIM 611560 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

The present paper focuses on literary creativity, and proposes a computational model for the creative process in this domain based on a number of cognitive model of the task of writing as carried out by humans.

2 Previous Work

The present paper puts forward a proposal that captures in computational terms the operations described in two existing cognitive models of the writing tasks. This section reviews these two models and two competing computational models also based them.

2.1 Cognitive Models of the Writing Task

Flower and Hayes [Flower and Hayes, 1981] define a cognitive model of writing in terms of three basic process: planning, translating these ideas into text, and reviewing the result with a view to improving it. These three processes are said to operate interactively, guided by a monitor that activates one or the other as needed. The planning process involves generating ideas, but also setting goals that can later be taken into account by all the other processes. The translating process involves putting ideas into words, and implies dealing with the restrictions and resources presented by the language to be employed. The reviewing process involves evaluating the text produced so far and revising it in accordance to the result of the evaluation. Flower and Hayes' model is oriented towards models of communicative composition (such as writing essays or functional texts), and it has little to say about literary creativity in particular. Nevertheless, a computational model of literary creativity would be better if it can be understood in terms compatible with this cognitive model. An important feature to be considered is that the complete model is framed by what Flower and Hayes consider "the rhetorical problem", constituted by the rhetorical situation, the audience and the writer's goals.

Sharples [Sharples, 1996] presents a description of writing understood as a problem-solving process where the writer is both a creative thinker and a designer of text. For Sharples, the universe of concepts to be explored in the domain of writing could be established in a generative way by exhaustively applying the rules of grammar that define the set of well-formed sentences. The conceptual space on which a writer operates is a subset of this universe identified by a set of constraints which define what is appropriate to the task at hand.

Sharples explains that the use of a conceptual space "eases the burden of writing by limiting the scope of search through long term memory to those concepts and schemas that are appropriate to the task" [Sharples, 1996, p. 3]. To Sharples, the imposition of these constraints enables creativity in the sense that he identifies creativity in writing (in contrast with simple novelty) with the application of processes that manipulate these constraints, thereby exploring and transforming the conceptual space that they define. Sharples provides a specification of what he envisages these constraints to be. Constraints on the writing task are described as "a combination of the given task, external resources, and the writer's knowledge and experience" [Sharples, 1996, p. 1]. He also mentions they can be external (essay topic, previously written material, a set of publishers guidelines...) or internal (schemas, inter-related concepts, genres, and knowledge of language that form the writer's conceptual spaces).

Sharples also provides a description of how the typical writer alternates between the simple task of exploring the conceptual space defined by a given set of constraints and the more complex task of modifying such constraints to transform the conceptual space. Sharples proposes a cyclic process moving through two different phases: engagement and reflection. During the engagement phase the constraints are taken as given and the conceptual space defined by them is simply explored, progressively generating new material. During the reflection phase, the generated material is revised and constraints may be transformed as a result of this revision. Sharples also provides a model of how the reflection phase may be analysed in terms of specific operations on the various elements. A three step process of reviewing, contemplating and planning the result is suggested as a description of the reflection phase. During reviewing the result is read, minor edits may be carried out, but most important it is interpreted to represent "the procedures enacted during composition as explicit knowledge which can then be integrated with an existing conceptual space". Contemplation involves the process of operating on the results of this interpretation, which are likely to be explicit representations of specific constraints. Planning uses the results of contemplation to create plans or intentions to guide the next phase of engagement.

Sharples also provides an account of how the explicit representation of constraints as elements susceptible of modification is fundamental to achieve this type of cyclic operation. People produce grammatically correct linguistic utterances without being aware of the rules of grammar, but to explore and transform conceptual spaces one needs to call up constraints and schemas as explicit entities, and work on them in a deliberate fashion. For the mind to be able to manipulate the constraints, they have to be subjected to a process of "representational redescription" [Karmiloff-Smith, 1995], re-representing knowledge that was previously embedded in effective procedures as elements susceptible of manipulation.

The problem is that beginners addressing such a cognitive task do not have a vocabulary to describe mental processes to themselves. To learn, they must develop "a coherent mental framework of plans, operators, genres and text types that can guide the process of knowledge integration and transformation" [Sharples, 1996, p. 5]. Experts tend to have such a mental framework that underlies and supports their writing efforts. For beginners, the problem must be addressed with the aid of general knowledge about how to design artefacts, how to transform mental structures and how to solve problems. Because this is difficult to do in the head, some writers resort to capturing the ideas involved in paper, as sketches, lists, plans, notes etc. These external representations stand for mental structures, and they are easier to manipulate. The writer can then explore different ways of structuring the content, apply systematic transformations, establish priorities, and reorder or cluster items. The task of writing addressed in these terms is much closer to recognised design tasks.

The arguments outlined above with respect to how Sharples models the differences between beginners and experts suggests further consideration of the role of the evolution of representation in the progressive acquisition of expertise. In this respect, Karmiloff-Smith [Karmiloff-Smith, 1995] proposes a model of evolving representation called Representational Redescription model.

This model analyses the development of behavioural mastery in a given domain – meaning consistenly successful performance in the domain – in terms of how knowledge about the domain is represented internally by the individual. The model considers three phases of learning. During the first phase the individual focuses on his interaction with the environment, and represents these in the form of raw data received from outside. This may lead to an initial achievement of behavioural mastery. Over the second phase, internal representations are abstracted from the raw data, and processing may start to focus on them. As a result of this introspection, features of the environment may temporarily be disregarded and, as a result, observed behaviour may deteriorate. However, this leads to a recuperation of a more flexible achievement of behavioural mastery, by then based on having achieved reconciliation between internal representation and external data.

This model describes four different levels of cognitive representation: *implicit*, focused on the process itself; *explicit level one* in which basic aggregation of raw data present in the implicit level is performed in terms of data storage but may not yet be accessible to the cognitive system for manipulation operations; *explicit level two*, in which structures from the first explicit level are converted into schemas and thereby become available; and *explicit level three*, a final and "cross-system" representation of concepts that can be verbalized and are fully integrated in a more general cognitive system.

2.2 Existing Implementations of Sharples' Model

MEXICA [Pérez y Pérez, 1999] was a computer model designed to study the creative process in writing in terms of the cycle of engagement and reflection [Sharples, 1999]. It was designed to generate short stories about the MEXICAS (also wrongly known as Aztecs). MEXICA is a flexible tool where the user can set the value of different parameters to constrain the writing process and explore different aspects of story generation. It takes into account emotional links and tensions between the characters as means for driving and evaluating ongoing stories.

MEXICA relies on certain structures to represent its knowledge: a set of *story actions* (defined in terms of pre-

conditions and post-conditions) and a set of *previous stories* (stated in terms of story actions). MEXICA stands out from other systems in that it actually builds its own set of schemas from the set of previous stories. A single type of knowledge structure, known as a *Story-World Context (SWC)*, is used to represent these schemas. Story-World Contexts represent instances of contexts (described in terms of emotional links and tensions between existing characters) in which an action has appeared in a previous story, and they act like rules during the engagement phase: an action is added to the plot if a Story-World Context for that action can be found that matches the plot so far.[1] The reflection phase revises the plot so far, mainly checking it for coherence, novelty and interest. The checks for novelty and interest involve comparing the plot so far with that of previous stories. If the story is too similar to some previous one, or if its measure of interest compares badly to previous stories, the system takes action by setting a guideline to be obeyed during engagement. These guidelines can be considered as a basic implementation of Sharples' constraints, driving which types of action can be chosen from the set of possible candidates.

In MEXICA, the system is actually aware of the emotions of all the characters (and the emotional tensions between them) and uses these to drive and structure the story. But these emotions and tensions are often not mentioned in the final text of the story.

2.3 The ICTIVS Model

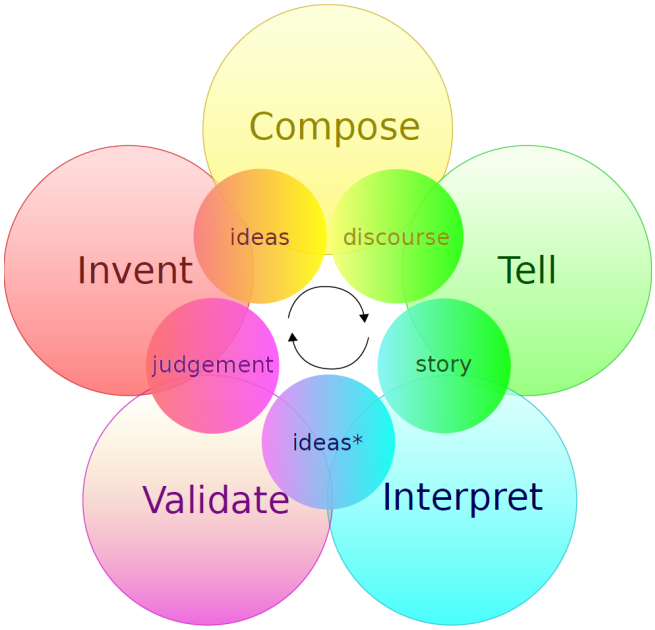


Figure 1: The original ICTIVS model. This version of the model does not take feedback into account.

The ICTIVS model [Gervás and León, 2014] arose as an initial attempt to construct a theoretical model based on an abstract analysis of the task of story construction in the context of a basic communication situation. Figure 1 shows a graphical representation of the ICTIVS model. The communication takes place as an exchange of a linear sequence of text that encodes a complex set of data that correspond to a set of events that take place over a volume of space time, possibly in simultaneous manner at more than one location. To convey this complexity as a linear sequence and recover it again at the other end of the communication process requires a process of condensing it first into a message and then expanding it again into a representation as close as possible to the original. There is a *composer*, in charge of composing a linear discourse from a conceptual source that may also have been produced by himself, and an *interpreter*, faced with the task of reconstructing a selected subset of the material in the conceptual source as an interpretation of the received narrative discourse. In real life, the role of the composer is usually played by a writer and the role of interpreter by a reader, but in the present case a more generic formulation has been preferred for generality.

This overarching act of communication is fundamental because it allows the definition of the purpose of the task in terms of the expected impact of the constructed story on the interpreter. Whatever is produced by the composer will have to be processed by the interpreter, and the impact on the interpreter cannot in truth be considered without taking into account what this process of interpretation involves. With this premise in mind, the ICTIVS model [Gervás and León, 2014] started from a linear description of the complete act of communication from an original purpose in the mind of the composer to a final impression in the mind of the interpreter. This act of communication involves processes of invention of a message and composition of an appropriate form meant to be carried out by the composer. It also involves processes of interpretation and validation carried out by the interpreter. From the point of view of the communicative act, the measure of success of such an act of needs to be established in terms of whether the interpretation by the interpreter matches the message constructed by the composer – success in terms of information transfer – and whether the impression in the mind of the interpreter matches the original purpose of the composer – success in terms of expected impact. As a first approximation, the impression in the mind of the interpreter could be correlated to the results of the validation applied to the message. In order to capture this intuition, the ICTIVS model defines the task of story composition as an iterative cycle of revisions in which the composer progressively generates drafts of his message, and then applies to them an internal process of interpretation and validation intended to match the one that the interpreter will be applying. At each iteration, the results of this estimated interpretation/validation are compared with the original purpose. If mismatches are detected, another cycle is started, and only when a successful match has been found does the resulting version of the message get communicated to the intended audience.

Five specific stages are included in the model: *INVENTION* – coming up with content for the narrative, possibly starting from scratch but often from some specification of

[1] It is important to note that Story-World Contexts (and not the definitions of action in terms of their pre-conditions) are used to find the next action to extend the plot.

purpose; a composer task –, *COMPOSITION* – establishing a form to express the desired content; a composer task –, *INTERPRETATION* – given a story, fill in the gaps, connect the dots, make assumptions on posible background implied, and extend it into a full picture of what the author wants you to "see in your mind"; an interpreter task that the composer needs to model to generate informative feedback for the construction process –, *VALIDATION* – identify the impact that the story, and/or the material interpreted from it, has on the interpreter; as above, an interpreter task but one that the composer needs to model to provide feedback –, and *TRANSMISSION* – passing over the result of the other processes to an audience; this stage establishes the link between the composer and the interpreter. Of these five stages, the first four may take place in an iterative cycle, and the final stage occurs only once after the iterations have lead to a successful draft with potential for achieving the expected impact on the interpreter according to the composer's purpose.

3 Reflective Feedback and Goal Revision in Computational Models of Literary Creativity

The computational models reviewed in section 2.2 and 2.3 capture the essence of the cognitive models described in section 2.1, but they both fail to capture the particular features that concern feedback and goal revision. The model of engagement and reflection in MEXICA has very limited explicit representation of the constraints driving the process, in the form of guidelines set during reflection. The ICTIVS model as originally described was formulated at a more abstract level, but focuses more on the constructive approach to the creative process, with no explicit modelling of the task of revising an already existing draft. It did include the representation of a seed idea or meaning that the composer wants to convey, but no representation of the possibility of this idea being modified as part of the creative process. A refinement on these models is required that can integrate a specification of the purpose for the generation task as an input, that can allow for revision of this specification as part of the process, and that at the same time can take advantage of the existing body of work on narrative generation.

Three relevant insights arise from the consideration of the original ICTIVS model in this enriched context of purpose-driven communication.

First, there will probably be a significant difference in computational terms between the initial iteration, where at each stage new material is generated from the corresponding input, and subsequent iterations, where two different processes may need to be employed: further generation of new material from the specification, and revision of the material generated in previous iterations - where the revision needs to be informed by the initial specification, the earlier drafts, and the identified mismatches. This is important because the computational mechanisms involved in each case may be different, and also because outputs from these two different processes may need to be combined into an integrated output for the corresponding stage.

Second, at the point of deciding whether a given draft is successful in terms of how it matches the original purpose, a truly creative process may consider not only revision of the draft but also revision of the purpose. This may arise whenever the estimated impact of a given draft on the interpreter is considered valuable by the composer beyond his original purpose. By means of this extension, the model can capture the role of serendipity in the creative process [Pease *et al.*, 2013; Corneli *et al.*, 2014].

Third, although computational models of the creative task are traditionally formulated as a cycle, in an ideal creative process cross-fertilization across the type of stages defined would be very positive. This is evident in Flower and Hayes description of the process as a set of transitions between three processes governed by an overall monitoring process that allocates effort to each one of them, and in Sharples' phrasing of his model as a dual cycle between two stages that operate on different data – the text and the constraints. A similar abstraction will need to be considered in our model.

The present section analyses these important concepts in more detail and attempts an initial formulation of such a refinement to take them into account in a manner that better reflects the intuitions arising from the cognitive models.

3.1 Analysing the Tasks Involved in Creative Production

In order to identify the core features that the desired model needs, the tasks involved in generation must be examined explicitly and compared to what the models can currently represent. Following this, we proposed a categorization of generative system according to their capabilities in terms of feedback.

Regarding their internal process, four types of systems can be identified:

- those that take no input and generate outputs determined exclusively by decisions taken during the construction of the system (*mere* generation)
- those that take as input some kind of specification that determines in some way the type of output that is to be obtained (*specification*)
- those that include a module that quantifies in some way the degree to which the outputs obtained satisfy the requirements specified as input (*diagnostic*)
- those that can benefit from the results of a diagnostic module to modify the specification and self correct their output (*reflective*)

Taking into account the kind of input that the systems accept, a parallel axis of classification may be whether the system can generate outputs only by constructing them from scratch (*construction*) or by applying transformations to an initial version of the desired artefacts (*revision*).

When these two axes are combined with the issues described previously, the following set of possible modes of operation arise:

- mere construction: the system generates outputs of a given form as determined by its construction
- construction to a specification: the system generates outputs conforming to a given specification

- construction with diagnostic: the system generates outputs and can provide some quantification of their quality
- reflective construction: the system generates outputs conforming to a given specification and can provide some quantification of their degree of satisfaction, and modify it accordingly.
- mere revision: the system receives an instance of the desired artefact and revises it towards a given goal determined by its construction
- revision to a specification: the system receives an instance of the desired artefact and a given specification and revises the instance of the desired artefact towards the given specification
- revision with diagnostic: the system receives an instance of the desired artefact, revises it towards a given goal determined by its construction, and can provide some quantification of the quality of the revised artefact
- reflective revision: the system receives an instance of the desired artefact and a given specification, revises the instance of the desired artefact towards the given specification, and can provide some quantification of their degree of satisfaction of the specification, and modify it accordingly.

3.2 Summarising the Features of a Creative Process from a Computational Point of View

The cognitive models reviewed in section 2.1 show a number of distinctive features that are relevant for the purpose of the present paper:

1. the creative process is iterative in nature
2. the creative process is driven by a set of constraints that restrict the desired outputs in some way; these constraints may be considered an input to the process
3. a cycle may involve processes of construction and/or processes of revision of prior results
4. at the end of each cycle a diagnostic procedure is applied to the result obtained so far
5. part of the diagnostic may involve quantifying degree of satisfaction of the given constraints
6. subsequent cycles take into account the diagnostic to attempt to improve the results of subsequent cycles
7. consideration of the diagnostic may take the form of planning further operations either on the artefact so far or on the set of constraints
8. the process as a whole includes a stage of meta-level reasoning which decides among the various available operational options applicable to the task at hand, such as, for instance, whether to iterate further or to stop, or, for a given iteration, whether to construct or to revise, whether to act upon the artefact itself or upon the set of constraints, or whether to apply the chosen operation to the whole element or to specific parts of it

3.3 Integrating the Reviewed Tasks and Features into a Computational Model

After having analysed both the tasks and the features involved in the creative process from a computational perspective, we propose the following three extensions for the refined model of the computations involved in literary creativity:

- to consider the explicit representation of constraints as part of the draft itself, so that they can be subject to the same operations as the rest of the draft
- to consider a range of operations that includes both construction and revision
- to consider the possibility of focusing system operation on particular subsets of the draft

The representation on which the creative process operates would therefore need to include at least two different parts:

- the set of constraints to be used to drive the construction process and/or to validate any resulting drafts, known as the *brief*
- the actual *draft* at each point of the creative process

Both the brief and the draft should be represented in such a way that different parts of them may be operated upon in isolation of the rest.

This representation that includes both a brief and a draft will be referred to henceforth as the *work in progress*. Any references to operations upon the work in progress can refer to both operations on the draft or on its specification.

The set of operations to consider would be:

- *reject*: eliminate from the work in progress a particular item for the next cycle
- *generate*: generate anew a particular item of the work in progress during the next cycle
- *revise*: modify a particular item of the work in progress during the next cycle
- *keep*: leave a particular item of the work in progress as it resulted from the previous cycle

Based on this terminology, a computational model for the tasks involved in literary creativity can now be rephrased at a lower level of detail. The same set of general steps can be seen, but each one of them now operates over a representation of the work in progress that includes both a brief and a draft, and at each stage the four types of operation (reject, generate, revise or keep) may be applied to any subset of the work in progress.

The computational model that we propose may now start from a hybrid representation of work in progress. Input may be provided to a creative system either in terms of a brief - a set of constraints that the output should satisfy – or a partial sample of the desired artefact, or as a combination of both modes.

This initial representation of the work in progress would undergo a process of reflection. In this initial reflection process, each of the sections of the work in progress is considered. If only a brief is available, the brief is marked as to be retained for the following construction cycle, and the empty

draft is tagged to be generated. If a brief and a partial draft are available, the partial draft is analysed in the light of the brief. The result of this analysis will be a diagnostic. Based on this diagnostic, the available draft is partitioned into sections, and each of these sections is marked as either to be left as it is (keep), to be regenerated (generate), to be revised (revise) or to be rejected (reject). Additionally, if the brief suggests sections should be added to the partial draft, place holders for them are added to the partial draft tagged as to be generated. If only a draft is available, an interpretation process is run on it to reverse engineer a brief. Based on the resulting brief, the available draft is processed as above. If neither brief nor draft are available, the creative system may follow different procedures, depending on whether a brief or a partial draft is constructed first.

Once the initial reflection phase is over, the system would enter a phase of construction. In spite of the similarities, we do not refer to this stage as engagement, because engagement in the sense used by Sharples applies very specifically to a process of production of new material, and the construction envisaged here may cover other processes such as revision, editing, or omisin. During this phase, each of the sections into which the draft has been partitioned will undergo the operation for which it has been tagged. The draft will therefore be edited by the application of the four basic operations described above. Any sections of the draft that are rejected at this point are stored in a log of fruitless paths.

At the end of the construction phase, the system would enter another phase of reflection. The first aim of this phase would be to ascertain whether the creative process has been concluded satisfactorily. This would arise if the draft matches the brief to perfection.

If the draft does not match the brief, the system would proceed with the rest of the reflection phase as described above, and iterated over another reflection/construction cycle. During reflection cycles other than the first one, the system may also consider modifications to the brief. These may arise from three possible situations. First, if part of the brief has proven impossible to satisfy during the prior cycle, the system may consider abandoning it. This would be plausible behaviour for human creators and should therefore be considered a possibility for artificial models. It would also constitute a very useful addition to allow creative system to steer themselves out of unproductive regions of a conceptual space when the current brief constrains them to restrict the search so. Second, if the reverse engineered brief shows positive features that were not included in the original brief, the system may decide to include them in the brief for the next iteration. This would allow such systems to incorporate the concept of serendipity into their computational models. Third, if the exploration of the conceptual space during a prior construction phase has included an excessive number of choices between possible candidate results, the system may decide to extend the brief to restrict the search to a subset of the conceptual space in question. Extensions to the brief should be compatible with the rest of it, and may take into account information about prior attempts that have failed.

Figure 2 depicts the reflective process in the proposed computational model for the tasks involved in literary creativity, as compared to the classic version that does not address feedback (depicted in Figure 1).

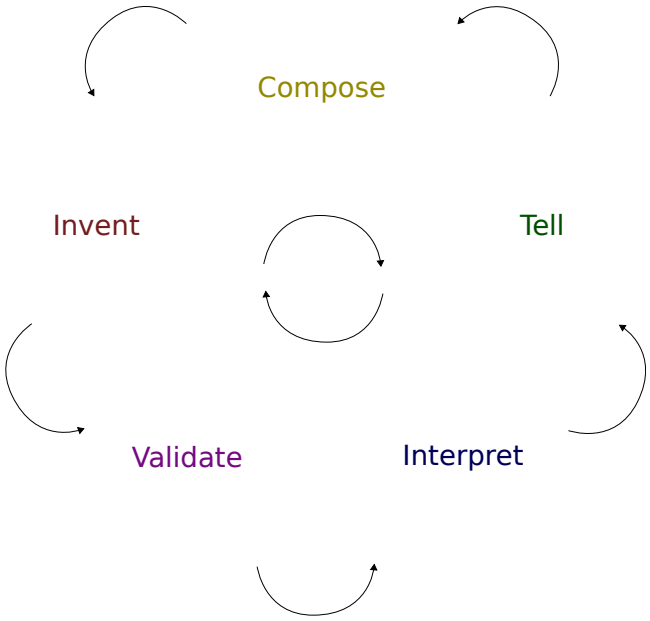


Figure 2: Graphical representation of the ICTIVS model as described in this paper, taking feedback into account. The feedback can trigger modifications of the brief (describing the constraints). The diagram represents draft generation in clockwise direction, and feedback revision in counter-clockwise direction.

4 Discussion

The planning process that Flower and Hayes has a dual purpose of generating ideas – which could correspond to additions to the ongoing draft – and of generating goals for the other processes. While the model that Flower and Hayes propose is not necessarily focused on creative writing, we consider it to be fundamental for describing narrative composition from the point of view of feedback. The ability to generate additions to the ongoing draft would correspond to Sharples process of engagement, with the slight refinement that the ideas generated in that case are restricted to the conceptual space defined by the initial constraints – which would correspond to our definition of the brief. These processes are covered by the generate option of our construction phase. The ability to set goals for other processes as established during the planning process that Flower and Hayes describe in their model corresponds to the establishment of constraints as described by Sharples. These processes are covered in our model by the part of the reflection stage where modifications to the brief are considered – in cycles beyond the initial one.

The translating process described by Flower and Hayes is described as a process of transforming ideas into text. According to our definitions, this would correspond to the task of generating a new instance of a particular section of the draft according to the corresponding brief, as carried out during a construction stage.

Our reflection process combines features of the reviewing process as described by Flower and Hayes – in as much as it involves evaluating the material produced so far and determining which parts of it can stand and which ones need further operations performed upon them – and the reflection stage of Sharples' model in the case of cycles beyond the initial one – where diagnostic leads not to revision of the draft but of the brief, which corresponds closely to Sharples' constraints. These constraints may then affect subsequent processes of revision but also of further construction or reconstruction of material already in the draft. The detailed description of the reflection stage as described by Sharples can be revisited using the proposed new terminology, which allows for finer consideration from a computational point of view of the actual operations involved at each point. The step that Sharples names reviewing involves a process of representational redescription – following Karmiloff-Smith – which clearly goes beyond comparing the results obtained with the original brief. Sharples specifically describes how the results obtained are processed to make available for reasoning within the system knowledge about "the procedures enacted during composition". Following Karmiloff-Smith, the raw data received from the generation processes are interpreted and internal representations are constructed describing valuable properties of these data at a more abstract level. This may take several forms, but a simple solution is to consider attempts to reverse engineer from the resulting draft a hypothetical brief that may have lead to it. This might be a reasonable match for the "explicit representations of specific constraints" that Sharples describes as likely outcomes of the process. The task of comparing such a reverse-engineered brief with the one actually used to drive the construction process would match Sharples step of contemplation, which operates on the results of the reviewing step. Sharples includes in the reflection stage an additional step of planning, where the results of contemplation lead to the creation of plans or intentions to guide the next phase of engagement. To fully capture the subtleties of Sharples analysis we have considered that the validation stage may result in the application of the four operations we have described – reject, generate, revise, keep – to the pair of briefs under consideration: the one used to generate the draft under revision and the one reverse engineered from the actual draft obtained. From this process a new brief will emerge, which can inherit constraints from the original brief, or delete them.

The reflection stage as we have described, in as much as it includes procedures for deciding which operations are to be carried out next on which parts of the work in progress, integrates the task of monitoring the creative process to guide the interaction and alternation between its constituent subprocesses – as described in Flower and Hayes model. The description provided in the present paper, given that it starts from a more fine-grained representation of both the data and the operations under consideration – allows for a more detailed and expressive description. This should allow for easier implementation of instances of creative systems that consider this type of behaviour.

With respect to the ICTIVS model, the model proposed in this paper may be seen as a refinement at a lower level of detail regarding the types of data involved and the types of operation carried out on them, but phrased at a higher level of abstraction with respect to the type of artefact being considered. The ICTIVS model was designed for the specific domain of narrative, and because of this it included separate stages for the ideation of plot or fabula and the composition of such plots or fabulae into sequential discourses. This would correspond to having different narrative levels of representation – fabula and discourse – for the material within the draft, and contemplating a specific process of conversion from one to the other. When we abstract away from these features specific to narrative, we can consider that the stages of *INVENTION* and *COMPOSITION* of the ICTIVS model would correspond to te construction phase that we have described in the current model, the *INTERPRETATION* and *VALIDATION* stage would correspond to the reflection stage, and the *TRANSMISSION* stage would correspond to the actual action of publishing or sending the final draft to an audience, which would correspond to the fulfilling the stopping condition implicit in our current phrasing of the reflection stage. With respect to the low level details described above, the task of reverse engineering a brief from a partial draft would match closely the processing that is considered during the interpretation phase of the ICTIVS model. The task of comparing such a reverse-engineered brief with the one actually used to drive the construction process would match the ICTIVS stage of validation.

The main advantage of this proposal is that two new sources of additional constraints are now included in the model of the creative process. First, the reverse-engineered brief may contain valuable constraints that were not in the original brief. This would correspond to the occurrence of serendipity: the constructive process employed leads to valuable features that were not in the original brief but which may be noticed during contemplation/interpretation of the result, and from then on added explicitly to the brief for subsequent iterations of the process. Second, the model allows for an explicit process for the generation of new constraints. This allows for the design of systems that can autonomously search for their own constraints, which would allow for a broader range of creative process. With respect to Boden's well known taxonomy of creative system [Boden, 2003], as the constraints being considered define the conceptual space that is being explored, a system capable of modifying the constraints that drive it might be capable of achieving transformational creativity. In this way the proposed models allows for a more fine grained representation of data and processes that may lead to the development of more expressive solutions.

The importance of making the system able to reject selected parts of its original brief should not be underestimated. The ability of human creators to depart – sometimes in very radical ways – from their original intentions in search for new aesthetics experiences has long been considered a critical ingredient of creativity of the highest order. It ties in very closely with the concept of transformational creativity, and the implicit ability to shift into new paradigms rather than just explore the old ones. Although all such issues are currently beyond the state of the art of creative systems, it is important to enable our computational models to represent

the types of behaviour that may one lead to implementation of similar behaviours. This would correspond to building explicitly into our computational models the idea of creativity at the metalevel [Wiggins, 2006]. The issue of how such operations might be profitably controlled is beyond the scope of the present paper and will need to be addressed in further work. Overall, a large proportion of the success of a creative system as described in the present proposal will depend on the implementation of suitable strategies for the partitioning of the draft into sections requiring the different operations available, and on the procedures for modifying the brief. These should be the focus of further work along the lines described in this paper.

5 Conclusions

The processes of literary creativity involve a complex web of interacting procedures (generation from a brief, evaluating how a draft matches a given brief, revision of an intermediate draft to fit a given brief, identifying unexpected valuable features from a working draft, editing a brief to optimise the search for creative results,...) and strategies for navigating between them. Existing cognitive models cover this space of solutions, but tend to remain at a high level of abstraction that leaves many of the features relevant for computation underspecified. The existing computational models of the writing task that have tried to take the cognitive models into account have focused on specific features of the process as their engineering mainstays, without trying to address the full complexity of the problem as a whole. The present paper proposes a computational model of the writing task that considers a broader set of ingredients than had been considered before, represented at a lower level of granularity in terms of their computational nature, both in terms of data and in terms of operations. The resulting model shows a strong potential for capturing significant phenomena in the field of creativity not often modelled computationally in the past, such as revision of drafts, working to a given brief, serendipity, and transformational creativity.

A valuable contribution of the proposed model is that it opens for exploration a significant number of lines of research to explore how these various phenomena might be addressed either in terms of working implementations of the proposed computational model or refinements of its basic formulation.

Acknowledgments

This paper has been partially supported by the project WHIM 611560 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

References

[Boden, 2003] Margaret Boden. *Creative Mind: Myths and Mechanisms*. Routledge, New York, NY, 10001, 2003.

[Corneli *et al.*, 2014] Joseph Corneli, Alison Pease, Simon Colton, Anna Jordanous, and Christian Guckelsberger. Modelling serendipity in a computational context. *CoRR*, abs/1411.0440, 2014.

[Flower and Hayes, 1981] L. Flower and J.R. Hayes. A cognitive process theory of writing. *College Composition and Communication*, 32(4):365–387, 1981.

[Gervás and León, 2014] P. Gervás and C. León. Reading and writing as a creative cycle: The need for a computational model. In *5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia, 06/2014 2014.

[Karmiloff-Smith, 1995] A. Karmiloff-Smith. *Beyond Modularity: A Developmental Perspective on Cognitive Science*. A Bradford book. A Bradford Book, 1995.

[Pease *et al.*, 2013] Alison Pease, Simon Colton, Ramin Ramezani, John Charnley, and Kate Reed. A discussion on serendipity in creative systems. In *Proceedings of the Fourth International Conference on Computational Creativity*, page 64–71, Sydney, Australia, jun 2013.

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

[Sharples, 1996] M. Sharples. An account of writing as creative design. In C. M. Levy and S. Ransdell, editors, *The Science of Writing: Theories, Methods, Individual Differences, and Applications*. Lawrence Erlbaum Associates, 1996.

[Sharples, 1999] M Sharples. *How We Write*. Routledge, 1999.

[Wiggins, 2006] G. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7), 2006.

Personalised Automated Assessments

Patricia Gutierrez and **Nardine Osman** and **Carles Sierra**
Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain
{patricia, nardine, sierra}@iiia.csic.es

Abstract

Consider an evaluator, or an assessor, who needs to assess a large amount of information. For instance, think of a tutor in a massive open online course with thousands of enrolled students, a senior program committee member in a large peer review process who needs to decide what are the final marks of reviewed papers, or a user in an e-commerce scenario where the user needs to build up its opinion about products evaluated by others. When assessing a large number of objects, sometimes it is simply unfeasible to evaluate them all and often one may need to rely on the opinions of others. In this paper we provide a model that uses peer assessments to generate expected assessments and tune them for a particular assessor. Furthermore, we are able to provide a measure of the uncertainty of our computed assessments and a ranking of the objects that should be assessed next in order to decrease the overall uncertainty of the calculated assessments.

1 Introduction

Consider an assessor who needs to assess a large amount of information. For instance, think of a tutor in a massive open online course with thousands of enrolled students, a senior program committee member in a large peer review process who needs to decide what are the final marks of reviewed papers, or a user in an e-commerce scenario where the user needs to build up its opinion about products evaluated by others. When assessing a large number of objects, sometimes it is simply unfeasible to evaluate them all and often one may need to rely on the opinions of others. In the process of building up our opinion, some questions need to be answered, such as: How much should I trust the opinion of a peer? What should I believe given a peer's opinion? What should I believe when many peers give their different opinions? Which objects should be assessed next, such that the certainty of my belief improves?

This paper addresses these questions through the Personalised Automated ASsessment model (PAAS). PAAS uses peer assessment to calculate and predict assessments. However, what is fundamentally different from many previous works [Piech *et al.*, 2013; de Alfaro and Shavlovsky, 2013; Walsh, 2014; Wu *et al.*, 2015] is that the computed peer-based assessment is tuned to the perspective of a specific community member. PAAS aggregates peer assessments giving more weight to those peers that are trusted by the specific community member whom the automated assessments are computed for. How much this specific member trusts a peer is then based on the similarity or evaluation rate between his (past) assessments and the peer's (past) assessments over the same assignments. To compute such a trust measure, we build a trust network conformed of direct and indirect trust values among community members. Direct trust values are derived from common assessments while indirect trust is based in the notion of transitivity. We clarify that our target is not consensus building, but to accurately estimate unknown assessments from a specific member's point of view, based on the peers' assessments and reliability.

Finally, we are also able to provide a measure of the uncertainty of our calculated assessments and a ranking of the objects that should be assessed next in order to decrease the overall uncertainty of those calculated assessments.

2 The PAAS Model

2.1 Notation and Problem Definition

Let ϵ represent an assessor who needs to assess a large set of objects $\mathcal{I}$, and let $\mathcal{P}$ be a set of peers that are able to assess objects in $\mathcal{I}$.

We understand assessments as probability distributions over an evaluation space $\mathcal{E}$ at a given moment in time. For example, one can define a set of elements for the evaluation space for the quality of an English classroom homework as $\mathcal{E} = \{poor, good, excellent\}$. The assessment $\{poor \mapsto 0, good \mapsto 0, excellent \mapsto 1\}$ would represent the highest assessment possible, whereas the assessment $\{poor \mapsto 0, good \mapsto 1/2, excellent \mapsto 1/2\}$ would represent that the quality of the homework is most probably between good and excellent, and so on.

We define an assessment e_i^α (also referred to as evaluation or opinion) as a probability distribution over the evaluation space $\mathcal{E}$, where $\alpha \in \mathcal{I}$ is the object being evaluated and $i \in \{\epsilon \cup \mathcal{P}\}$ is the evaluator. We say $e_i^\alpha = \{x_1 \mapsto v_1, \ldots, x_n \mapsto v_n\}$, where $\{x_1, \ldots, x_n\} = \mathcal{E}$ and $v_i \in [0, 1]$ represents the value assigned to each element $x_i \in \mathcal{E}$, with the condition

that $\sum_{i \in |\mathcal{E}|} v_i = 1$.

Finally, we define $\mathcal{L}$ as the history of all assessments performed, and $\mathcal{O}_\alpha \subset \mathcal{L}$ as the set of past peer assessments over the object α.

The ultimate goal of our work is to compute the probability distribution of ϵ's evaluation over a certain object α, given the evaluations of several peers over that same object α. In other words, what is the probability that ϵ's evaluation is x given the set of peers' evaluations O_α? Such expectation can be formalized with the conditional probability as follows:

$$p(X{=}x \mid \mathcal{O}_\alpha)$$

.

To calculate the above conditional probability, we take into account every particular evaluation in O_α. In other words, expectations (or probabilities) are calculated for each individual evaluation in O_α, before those expectations are aggregated into $p(X{=}x \mid \mathcal{O}_\alpha)$. The probability that ϵ's assessment is x given a particular evaluation $e_\mu^\alpha \in O_\alpha$ is formalized as follows:

$$p(X{=}x \mid e_\mu^\alpha)$$

.

The more general probability $p(X{=}x \mid \mathcal{O}_\alpha)$ is then defined as an aggregation of the individual probabilities:

$$p(X{=}x \mid \mathcal{O}_\alpha){=}\overline{p(X{=}x \mid e_\mu^\alpha)}$$

where the exact definition of the aggregation is presented later on in Section 2.4.

We strongly base the intuition behind the computation of the individual conditional probabilities on the notion of *trust* between peers based on previous experiences, where trust is understood in this context as the expected similarity between the assessments given by those peers. In other words, our intuition is that we expect ϵ will tend to agree with μ's assessments if his trust on μ is high. Otherwise, ϵ's evaluation will probably be different. We perform then a sort of analogical reasoning: if in the past μ gave opinions that were a certain degree dissimilar from ϵ's opinions, then this will probably happen again now.

The remainder of this section is divided accordingly. We first describe in detail how the measure of trust between peers is calculated (Section 2.2). Then, we illustrate how to calculate ϵ's assessment on an object α given μ's assessment over α and ϵ's trust in μ's assessments (Section 2.3). In other words, we present an approach for calculating the individual probability $p(X{=}x \mid e_\mu^\alpha)$. We then illustrate how to combine those probabilities to build the probability distribution of ϵ's assessments given the assessments of several peers (Section 2.4). In other words, we present an approach for calculating the probability $p(X{=}x \mid \mathcal{O}_\alpha)$. Finally, we provide a measure of the uncertainty of the computed assessments and a ranking of the objects that should be assessed next by ϵ in order to decrease that uncertainty (Section 2.5).

2.2 Step 1. How much should I trust a peer?

ϵ needs to decide how much can he or she trust the assessment of a peer μ. We define this trust measure based on the following two intuitions. Our first intuition states that if ϵ and μ have both assessed the same object, then the similarity of their assessments can give a hint of how close their judgments are. However, cases may arise where there are simply no objects evaluated by both ϵ and μ. In such a case, one may think of simply neglecting μ's assessment, as ϵ would not know how much to trust μ's assessment. Our second intuition, however, proposes an alternative approach for such cases, where we approximate that unknown trust between ϵ and μ by looking into a chain of trust between ϵ and μ through other peers. Roughly speaking, we relay on the transitive notion: "if ϵ trusts μ, and μ trusts μ', then ϵ will likely trust μ'". In the following, we define these two intuitions through two different types of trust relations: direct trust and indirect trust.

Direct Trust

Direct trust is the trust relation that emerges between evaluators that have assessed one or more objects in common. One possible approach is to measure such relation as aggregations of their evaluations' similarity over those objects assessed in common. For instance, let the set $A_{i,j}{=}\{\alpha \mid e_i^\alpha, e_j^\alpha \in \mathcal{L}\}$ be the set of objects that have been assessed by both evaluators i and j. Then different definitions for the direct trust between i and j based on the similarity between two assessments ($sim(e_j^\alpha, e_j^\alpha)$) may be adopted, such as as:

- The average of the similarities for all commonly assessed objects:

$$T_D(i,j){=}\frac{\sum_{\alpha \in A_{i,j}} sim(e_i^\alpha, e_j^\alpha)}{|A_{i,j}|}$$

- The conjunction of the similarities for all commonly assessed objects:

$$T_D(i,j){=}\bigwedge_{\alpha \in A_{i,j}} sim(e_i^\alpha, e_j^\alpha)$$

- The Pearson coefficient [Upton and Cook, 2008], or linear correlation between i and j, for all commonly assessed objects:

$$T_D(i,j){=}\frac{\sum_{\alpha \in A_{i,j}} sim(e_i^\alpha, \bar{e_i}) \cdot sim(e_j^\alpha, \bar{e_j})}{\sqrt{\sum_{\alpha \in A_{i,j}} sim(e_i^\alpha, \bar{e_i})^2} \sqrt{\sum_{\alpha \in A_{i,j}} sim(e_j^\alpha, \bar{e_j})^2}}$$

  where $\bar{e_i}$, $\bar{e_j}$ are the means of the evaluations performed over the set $A_{i,j}$ by i and j respectively.

However when we calculate such aggregations we loose relevant information. For instance, we are not able to tell if j usually under rates with respect to i, if it usually over rates, or neither. We are also not able to tell if the dissimilarities between i and j's evaluations are highly variable or not.

To cope with such loss of information, we define the direct trust between two peers i and j as a probability distribution

$\mathbb{T}_{\mathbb{D}i,j} : [0,1] \rightarrow [0,1]$ built from the historical data of previous evaluations performed by i and j. This probability distribution describes, as we will explain shortly, the *expected similarity* or the *expected evaluation rate* between i and j's assessments. The support of the distribution is $[0,1]$ since both the expected similarity and the expected evaluation rate are in the range $[0,1]$, as we will see shortly, and the range of the distribution is $[0,1]$ as this is a probability distribution and the range of any probability is $[0,1]$. Note that we do not consider here any summarizing measure for trust that would translate that distribution into a single value, although a number of measures could be used, such as the average similarity (as the center of gravity of the distribution) or entropy (as a measure of the uncertainty of the distribution).

When defining $\mathbb{T}_{\mathbb{D}i,j}$ we distinguish two cases: (1) a first case with a non-ordered evaluation space, such as $\mathcal{E} = \{visionary, original, sound\}$; and (2) a second case with an ordered evaluation space, such as $=\{bad, good, excellent\}$. In the second case, we are interested in maintaining information about whether a peer under rates or over rates with respect to another peer, therefore we are interested in the *expected evaluation rate* between i and j. In the first case, this is not an issue as assessments cannot be ordered and therefore the notion of under/over rating does not exist, therefore we are rather interested in the *expected similarity* between i and j's assessments. Next we detail the trust probability distributions $\mathbb{T}_{\mathbb{D}i,j}$ built for both cases.

- *Non-Ordered Case*.

  In the non-ordered case, we are interested in the similarity between i and j's assessments. As such, the support of the distribution representing i's direct trust on j (i.e. the x-axis of $\mathbb{T}_{\mathbb{D}i,j}$) consists of the possible degrees of similarity between i and j's assessments.

  Trust distribution $\mathbb{T}_{\mathbb{D}i,j}(x)$ then describes the probability that peers i and j evaluate an object with a similarity x (or the probability that the similarity of their evaluations is x).

- *Ordered Case.*

  In the ordered case, we are interested in the evaluation rate ${}^{e_j}/_{e_i}$ between evaluations made by peers i and j. If ${}^{e_j}/_{e_i} = 1$, this means that i and j provide the same evaluation. If ${}^{e_j}/_{e_i} > 1$, this meas that j over rates with respect to i. If ${}^{e_j}/_{e_i} < 1$, this means that j under rates with respect to i.

  We normalize the evaluation rate to values between 0 and 1. To do so, we require a non decreasing function $r : \mathcal{R} \rightarrow [0,1]$ such that $\lim_{x\rightarrow\infty} r(x)=1$, and for convenience we constraint $r(1)=0.5$. We adopt the following normalized evaluation rate function that satisfies these properties:

  $$r(x)=e^{\ln {}^1/_{2/x}} \quad (1)$$

  As such, the support of the distribution representing i's direct trust on j (i.e. the x-axis of $\mathbb{T}_{\mathbb{D}i,j}$) consists of the possible normalized evaluation rates between i and j. Trust distribution $\mathbb{T}_{\mathbb{D}i,j}(x)$ then describes the probability that i and j would assess an object with a normalized evaluation rate x.

In what follows, we explain how we build direct trust distributions computationally, based on previous experiences.

Initially, the direct trust distribution between any two peers is the uniform distribution $\mathbb{F}=\{{}^1/_n, \ldots, {}^1/_n\}$ (describing ignorance), where n is the size of the distribution's support. Every new assessment made would then update the trust distributions accordingly. Consider a new assessment e_i^α. The distribution $\mathbb{T}_{\mathbb{D}i,j} \forall j$ s.t. $A_{i,j} \neq \emptyset$ is updated as follows:

1. We find the element x in $\mathbb{T}_{\mathbb{D}i,j}$'s support whose probability needs to be adjusted. So we calculate $x{=}sim(e_j^\alpha, e_i^\alpha)$ in the ordered case (where the definition of sim is domain dependent and outside the scope of this paper, although we do note that several approaches may be adopted, such as using semantic similarity measures [Li *et al.*, 2003]), or $x = r({}^{e_j^\alpha}/_{e_i^\alpha})$ in the non-ordered case (Equation 1).

2. We update the probability of the *single expectation* x in $\mathbb{T}_{\mathbb{D}i,j}$ accordingly:

   $$p(X{=}x) = p(X{=}x) + \gamma \cdot (1 - p(X{=}x)) \quad (2)$$

   The update is based on increasing the latest probability $p(X{=}x)$ by a fraction $\gamma \in [0,1]$ of the total potential increase $(1 - p(X{=}x))$. For instance, if the probability of x is 0.6 and γ is 0.1, then the new probability of x becomes $0.6 + 0.1 \cdot (1 - 0.6) = 0.64$. We note that the ideal value of γ should be closer to 0 than to 1 so that one single experience does not result in considerable changes in the distribution. In other words, a *single* assessment cannot result in *considerable* change in the probability distribution. Considerable changes can only be the result of information learned from the accumulation of many assessments.

3. We normalize $\mathbb{T}_{\mathbb{D}i,j}$ by updating *several expectations* following the entropy based approach of [Sierra and Debenham, 2005]. The entropy-based approach updates $\mathbb{T}_{\mathbb{D}i,j}$ such that: (1) the value $p(X{=}x)$ is maintained and (2) the resulting distribution has a minimal relative entropy with respect to the previous one. In other words, we look for a distribution that contains the updated probability value $p(X{=}x)$ and that is at a minimal distance from the original $\mathbb{T}_{\mathbb{D}i,j}$ (as the relative entropy is a measure of the difference between two probability distributions). Following this approach, we update $\mathbb{T}_{\mathbb{D}i,j}(X)$ as follows:

   $$\mathbb{T}_{\mathbb{D}i,j}(X) = \underset{\mathbb{P}'(X)}{\arg\min} \sum_{x'} p(X{=}x') \log \frac{p(X{=}x')}{p'(X{=}x')} \quad \text{such that} \quad \{p(X{=}x) = p'(X{=}x)\} \quad (3)$$

   where $p(X{=}x')$ is a probability value in $\mathbb{T}_{\mathbb{D}i,j}$, $p'(X{=}x')$ is a probability value in $\mathbb{P}'$, and $\{p(X{=}x) = p'(X{=}x)\}$ specifies the constraint that needs to be satisfied by the resulting distribution.

Indirect Trust

Given a direct trust relation between peers i and j and between peers j and k, the question now is: What can we say about the indirect trust between peers i and k when i and k

have no objects assessed in common? In other words, given the direct trust distributions $\mathbb{T}_{\mathbb{D}i,j}$ and $\mathbb{T}_{\mathbb{D}j,k}$, what can we say about the indirect trust distribution $\mathbb{T}_{\mathbb{I}i,k}$?

As with direct trust distributions, we distinguish two cases: a first case where assessments cannot be ordered and thus trust is based on a similarity measure sim; and a second case where assessments can be ordered and thus trust is based on a normalized evaluation rate function $r(x)=e^{\ln 1/2/x}$.

- *Non-Ordered Case*.

  In this case, we want to preserve the fundamental triangular inequality property of similarity functions that says that: T-norm$(sim(a,b), sim(b,c)) \leq sim(a,c)$. As with $\mathbb{T}_{\mathbb{D}i,k}$, the support (or the x-axis) of $\mathbb{T}_{\mathbb{I}i,k}$ consists of the possible degrees of similarity between i and k's assessments. But since these degrees of similarity should satisfy the T-norm, the support is defined as the set:

  $$\text{supp}(\mathbb{T}_{\mathbb{I}i,k})=\{x_{ik}=\text{T-norm}(x_{ij},x_{jk}) \mid x_{ij} \in \text{supp}(\mathbb{T}_{\mathbb{D}i,j}) \wedge x_{jk} \in \text{supp}(\mathbb{T}_{\mathbb{D}j,k})\}$$

  where supp represents the support of a distribution.

  We then compute the probabilities of the expectations of $\mathbb{T}_{\mathbb{I}i,k}$ as follows:

  $$\{p(X=x_{ik}=\text{T-norm}(x_{ij},x_{jk}))=\mathbb{T}_{\mathbb{D}i,j}(x_{ij}) * \mathbb{T}_{\mathbb{D}j,k}(x_{jk}) \mid x_{ij} \in \text{supp}(\mathbb{T}_{\mathbb{D}i,j}) \wedge x_{jk} \in \text{supp}(\mathbb{T}_{\mathbb{D}j,k})\} \quad (4)$$

  This could result in more than one probability computed for the same expectation x_{ik}. As such, we then add up all the probabilities that correspond to the same expectation x_{ik}.

  We note that we follow a conservative approach by adopting the product operator (Equation 4), which is a T-norm that gives the smallest possible values, as we prefer not to overrate indirect trust values since they are not inferred directly from historical data. Of course, other operators could also be used, such as the min function.

- *Ordered Case*.

  In this case, we want to preserve the property: $e_j/e_i * e_k/e_j = e_k/e_i$ with respect to the evaluations performed by i, j and k. For instance, if the evaluation rate between e_j and e_i is 0.5 (j under rates a 50% with respect to i) and the evaluation rate between e_k and e_j is 0.5 (k under rates a 50 % with respect to j) then the evaluation rate between e_k and e_i should be 0.25 (then k under rates a 75 % with respect to i).

  As above, the support (or the x-axis) of $\mathbb{T}_{\mathbb{I}i,k}$ consists of the possible degrees of similarity between i and k's assessments. The support us then defined as the set:

  $$\text{supp}(\mathbb{T}_{\mathbb{I}i,k}) = \{x_{ik}=x_{ij} * x_{jk} \mid x_{ij} \in \text{supp}(\mathbb{T}_{\mathbb{D}i,j}) \wedge x_{jk} \in \text{supp}(\mathbb{T}_{\mathbb{D}j,k})\}$$

  We then compute the probabilities of the expectations of $\mathbb{T}_{\mathbb{I}i,k}$ as follows:

  $$\{p(X=x_{ik}=x_{ij} * x_{jk}) = \mathbb{T}_{\mathbb{D}i,j}(x_{ij}) * \mathbb{T}_{\mathbb{D}j,k}(x_{jk}) \mid x_{ij} \in \text{supp}(\mathbb{T}_{\mathbb{D}i,j}) \wedge x_{jk} \in \text{supp}(\mathbb{T}_{\mathbb{D}j,k})\} \quad (5)$$

  Again, this could result in more than one probability computed for the same expectation x_{ik}. As such, we then add up all the probabilities that correspond to the same expectation x_{ik}.

The calculations presented above provide an approach for calculating indirect trust between two peers i and k when those peers are linked through a direct trust chain passing through only one intermediate peer j. For direct trust chains of increasing length between i and k, the previous process is iterated. For instance, if there is a direct trust chain linking i to j, j to m, and m to k, then we first compute the indirect trust distribution $\mathbb{T}_{\mathbb{I}i,m}$ from the direct trust distributions $\mathbb{T}_{\mathbb{D}i,j}$ and $\mathbb{T}_{\mathbb{D}j,m}$, and then we compute the indirect trust distribution $\mathbb{T}_{\mathbb{I}i,k}$ from the direct/indirect trust distributions $\mathbb{T}_{\mathbb{I}i,m}$ and $\mathbb{T}_{\mathbb{D}m,k}$, following the same approach as above.

When multiple chains of direct trust connect two peers (e.g. say a chain linking i to j and j to k, and another chain linking i to m and m to k), we obtain multiple indirect trust distributions (one from every chain). In those cases, we pick the resulting distribution which is most optimistic. In other words, while our approach to calculate the indirect trust follows the pessimistic approach (through our choice of the product operator in Equations 4 and 5), we now choose the most optimistic of the pessimistic outcomes. To do that, we choose the distribution that is closest to the *equivalence* distribution, which is a distribution that describes that the evaluations of two peers are equivalent. In the non-ordered case, the equivalence distribution is $\mathbb{P}_{\mathbb{E}}(1)=1$; that is, the similarity between two peers is maximum. In the non-ordered case, the equivalence distribution is $\mathbb{P}_{\mathbb{E}}(0.5) = 1$; that is, the normalized evaluation rate between two peers is 0.5, which implies that they always provide the same evaluation. The distance between an indirect trust distribution $\mathbb{T}_{\mathbb{I}i,k}$ and the equivalence distribution $\mathbb{P}_{\mathbb{E}}$ can be calculated as:

$$emd(\mathbb{T}_{\mathbb{I}i,k}, \mathbb{P}_{\mathbb{E}}) \quad (6)$$

where emd is the earth mover's distance which calculates the distance between two probability distributions [Rubner *et al.*, 1998].[1] We note that the range of emd is [0,1], where 0 represents the minimum distance and 1 represents the maximum possible distance.

In the remainder of this paper, when we refer explicitly to a direct or indirect trust distribution between peers i and j, we refer to such distribution as $\mathbb{T}_{\mathbb{D}i,j}$ or $\mathbb{T}_{\mathbb{I}i,j}$, respectively. Whereas when we refer generically to a trust distribution that could either be the direct or indirect trust distribution, we refer to such a distribution as $\mathbb{T}_{i,j}$.

Trust Graph

Direct and indirect trust relations in a community can be represented by a weighted directed graph. We define a community's *trust graph* as:

$$G=\langle N, E, w\rangle$$

[1] If probability distributions are viewed as piles of dirt, then the earth mover's distance measures the minimum cost for transforming one pile into the other. This cost is equivalent to the 'amount of dirt' times the distance by which it is moved, or the distance between elements of the probability distribution's support.

where the set of nodes N is the set of evaluators in $\{\epsilon \cup \mathcal{P}\}$, $E \subseteq N \times N$ are edges between evaluators with direct or indirect trust relations, and $w : E \mapsto [0,1]^n$ is the weight of an edge, described as a trust probability distribution.

$D \subset E$ is the set of edges that link evaluators with direct trust relations: $D = \{(i,j) \in E \mid \mathbb{T}_{\mathbb{D}i,j} \neq \perp\}$. Similarly, $I \subset E$ is the set of edges that connect evaluators with indirect trust relations: $I = \{(i,j) \in E \mid \mathbb{T}_{\mathbb{I}i,j} \neq \perp\} \setminus D$. We note that the set of edges E is then composed of the union of the set of direct and indirect edges: $E = D \cup I$. Weights in w describe direct and indirect trust probability distributions and are defined as follows:

$$w(i,j) = \begin{cases} \mathbb{T}_{\mathbb{D}i,j} & \text{, if } (i,j) \in D \\ \mathbb{T}_{\mathbb{I}i,j} & \text{, if } (i,j) \in I \end{cases}$$

Our goal is to determine how much a particular evaluator ϵ can trust a peer μ. So the trust graph is constructed with respect to ϵ's point of view only. Therefore, we maintain a trust graph of the whole community containing all the *direct* edges between peers (as they are needed to calculate indirect trust relations), but we only maintain the *indirect* edges that connect ϵ with the rest of the peers.

Information Decay

An important notion in our proposal is the notion of the *decay* of information. We say the integrity of information decreases with time. In other words, the information provided by a trust probability distribution should lose its value over time and decay towards a default value. We refer to this default value as the *decay limit distribution* $\mathbb{D}$. For instance, $\mathbb{D}$ may be the uniform distribution, which describes that trust information learned from past experiences tends to ignorance over time.

To implement such a decay mechanism, we need to:

1. Record all evaluations $e_i^\alpha \in \mathcal{L}$ made at time t with a timestamp t, noted $e_i^{\alpha^t}$.
2. Record all direct trust distributions $\mathbb{T}_{\mathbb{D}i,j}$ with a timestamp t, noted $\mathbb{T}_{\mathbb{D}i,j}^t$, where t is the timestamp of the last evaluation that modified the trust distribution. The first time $\mathbb{T}_{\mathbb{D}i,j}$ is calculated, t is the timestamp of the latest evaluation amongst the two evaluations leading to this calculation. (Recall that it is the similarity between two evaluations or the evaluation rate that updates the probability distribution.) Then, every time a new evaluation with timestamp $t' > t$ is considered to update $\mathbb{T}_{\mathbb{D}i,j}^t$, $\mathbb{T}_{\mathbb{D}i,j}^t$ is first decayed from t to t' before the distribution is updated.
3. Record all indirect trust distributions $\mathbb{T}_{\mathbb{I}i,j}$ with a timestamp t, noted $\mathbb{T}_{\mathbb{I}i,j}^t$, where t is the time the distribution is calculated. Every time $\mathbb{T}_{\mathbb{I}i,j}$ is calculated, all probability distributions involved in this calculation will first need to be decayed to the time of calculation t. The time of calculation is usually the latest timestamp amongst the timestamps of the distributions involved in this calculation.

Information in a trust probability distribution $\mathbb{T}_{i,j}$ decays from t to t' (where $t' > t$) as follows:

$$\mathbb{T}_{i,j}^{t \rightsquigarrow t'} = \Lambda\ (\mathbb{D}, \mathbb{T}_{i,j}^t) \tag{7}$$

where Λ is the *decay function* satisfying the property: $\lim_{t' \to \infty} \mathbb{T}_{i,j}^{t \rightsquigarrow t'} = \mathbb{D}$. One possible definition for Λ could be:

$$\mathbb{T}_{i,j}^{t \rightsquigarrow t'} = \nu^{\Delta_{t,t'}} \cdot \mathbb{T}_{i,j}^t + (1 - \nu^{\Delta_{t,t'}})\mathbb{D} \tag{8}$$

where ν is the decay rate, and:

$$\Delta_{t,t'} = \begin{cases} 0 & \text{, if } t' - t < \omega \\ 1 + \dfrac{t' - t}{t_{max}} & \text{, otherwise} \end{cases}$$

The definition of $\Delta_{t,t'}$ above serves the purpose of establishing a minimum *grace* period, determined by the parameter ω, during which the information does not decay, and that once reached the information starts decaying. The parameter t_{max}, which may be defined in terms of multiples of ω, controls the *pace of decay*. The main idea behind this is that after the grace period, the decay happens very slowly; in other words, $\Delta_{t,t'}$ decreases very slowly.

2.3 Step 2: What to belief when a peer gives an opinion?

Given a peer assessment e_μ^α, the question now is how to compute the probability distribution of ϵ's evaluation. In other words, what is the probability that ϵ's evaluation of α is x given that μ evaluated α with e_μ^α. As illustrated earlier, this is expressed as the conditional probability:

$$\mathbb{P}(X{=}x \mid e_\mu^\alpha)$$

To calculate this conditional probability, the intuition is that ϵ would tend to agree with μ's evaluation if his trust on μ (that is, the expected similarity between their assessments or the expected evaluation rate between their assessments) is high. Otherwise, ϵ's evaluation would probably be different. We perform then a sort of analogical reasoning: if in the past μ gave assessments that were a certain degree dissimilar from ϵ's opinions, or with a certain evaluation rate with respect to ϵ, then this will probably happen again now.

We then calculate the above conditional probability based on the following desired properties:

- If $\mathbb{T}_{\epsilon,\mu}$ is a flat distribution (i.e. a distribution representing ignorance), then $\mathbb{P}(X \mid e_\mu^\alpha)$ should also be a flat distribution. That is, the closer ϵ's trust on μ is to ignorance, the less information μ is giving to ϵ with his/her assessment.
- The degree of belief $e_\epsilon^\alpha = x$ should increase for those points x whose similarity (or evaluation rate, in the case of the ordered case) to e_μ^α is high (i.e. for higher values of $\mathbb{T}_{\epsilon,\mu}$).
- The degree of belief $e_\epsilon^\alpha = x$ should decrease for those points x whose similarity (or evaluation rate, in the case of the ordered case) to e_μ^α is low trust (i.e. for lower values of $\mathbb{T}_{\epsilon,\mu}$).

Formally, these properties are achieved by defining the probabilities accordingly (where the denominator of the following two equations, Equations 9 and 10, is used for normalisation to ensure that the resulting distribution is a probability distribution):

- *Non-Ordered Case.*

$$p(X{=}x \mid e_\mu^\alpha) = \frac{e^{\mathbb{T}_{\epsilon,\mu}(sim(e_\mu^\alpha,x))\cdot\mathbb{I}(\mathbb{T}_{\epsilon,\mu})}}{\sum_{x'\in\mathcal{E}} e^{\mathbb{T}_{\epsilon,\mu}(sim(e_\mu^\alpha,x'))\cdot\mathbb{I}(\mathbb{T}_{\epsilon,\mu})}} \quad (9)$$

- *Ordered Case.*

$$p(X{=}x \mid e_\mu^\alpha) = \frac{e^{\mathbb{T}_{\epsilon,\mu}(r(e_\mu^\alpha/x))\cdot\mathbb{I}(\mathbb{T}_{\epsilon,\mu})}}{\sum_{x'\in\mathcal{E}} e^{\mathbb{T}_{\epsilon,\mu}(r(e_\mu^\alpha/x'))\cdot\mathbb{I}(\mathbb{T}_{\epsilon,\mu})}} \quad (10)$$

where $\mathbb{I}(\mathbb{T}_{\epsilon,\mu})$ is a measure of how informative the probability distribution $\mathbb{T}_{\epsilon,\mu}$ is. We calculate $\mathbb{I}(\mathbb{T}_{\epsilon,\mu})$ as:

$$\mathbb{I}(\mathbb{T}_{\epsilon,\mu}) = 1 - \mathbb{H}(\mathbb{T}_{\epsilon,\mu}) \quad (11)$$

where $\mathbb{H}$ describes the entropy of a probability distribution. In other words, the lower the entropy of the distributions then the more informative it is, and vice versa.

We finally define the probability distribution of ϵ's expected evaluation given μ's opinion accordingly: $\mathbb{P}(X \mid e_\mu^\alpha)$, where X varies over the evaluation space $\mathcal{E}$.

2.4 Step 3: What to belief when many give opinions?

In the previous section we computed $\mathbb{P}(X \mid e_\mu^\alpha)$. That is, the probability distribution of ϵ's evaluation on α given the evaluation of a peer μ on α. But what does ϵ do when there is more than one peer assessing α?

Given the set of opinions $\mathcal{O}_\alpha$ describing a set of peer evaluations over the object α, we define the probability of ϵ's assessment being x as follows:

$$p(X{=}x \mid \mathcal{O}_\alpha)) = \frac{\prod_{\mu\in O_\alpha} p(X{=}x \mid e_\mu^\alpha)}{\sum_{x'\in\mathcal{E}} \prod_{\mu\in\mathcal{O}_\alpha} p(X{=}x' \mid e_\mu^\alpha)} \quad (12)$$

In other words, the probability of ϵ's assessment on α being x given the set of opinions over α is an aggregation (a product in this case) of the probabilities of ϵ's assessment on α being x given each evaluation $e_\mu^\alpha \in \mathcal{O}_\alpha$.

We then define the probability distribution of ϵ's expected evaluation given all opinions in $\mathcal{O}_\alpha$ as $\mathbb{P}(X \mid \mathcal{O}_\alpha)$, where X varies over the evaluation space $\mathcal{E}$.

We note that instead of the product operator $\prod$ other connectives could be used, for instance the $\min$ operator might be used. However, we note that using the minimum operator does not take into account the number of assessments made. That is, having assessments of 20 peers could be equivalent to having the assessment of just one peer. In fact, the proposed aggregation of Equation 12 ensures that:

- The larger the number of identical opinions, the less uncertain the final probability distribution is, and
- The more trusted the opinions, the less uncertain the final probability distribution is.

Finally, to translate the final assessment from a probability distribution $\mathbb{P}(X \mid \mathcal{O}_\alpha)$ into a single value, we calculate the mean (average) of the distribution and select the closest mark to that mean.

2.5 Step 4: What should be evaluated next?

The previous three steps have provided a model to calculate automated assessments of objects that have not been assessed by ϵ, based on peers opinions. The level of uncertainty of the automated assessments generated by our model can be calculated as the uncertainty of the probability distribution of ϵ's expected evaluation based on those peers opinions $\mathbb{P}(X \mid \mathcal{O}_\alpha)$. This level of uncertainty is measured by the distribution's entropy:

$$\mathbb{H}(\mathbb{P}(X \mid \mathcal{O}_\alpha))$$

The question that naturally arises then is what objects can be assessed next by ϵ to decrease such uncertainties? For example, how many more assignments should a tutor evaluate so that the uncertainty of the calculated assessments becomes *acceptable*. We suggest ϵ to evaluate objects with maximum uncertainty, or maximum entropic value. The ranking of objects with respect to their entropic value is then defined as follows:

$$\begin{aligned} Rank(\alpha) &= 1 - \mathbb{H}(\mathbb{P}(X \mid \mathcal{O}_\alpha)) \\ &= 1 + \sum_{x\in X} p(X{=}x \mid \mathcal{O}_\alpha) \ln p(x \mid \mathcal{O}_\alpha) \end{aligned} \quad (13)$$

ϵ can then continue to evaluate objects one by one until the uncertainty of the automated assessments becomes less than some predefined *acceptable uncertainty threshold*.

3 Conclusions and Future Work

In this paper we have presented the personalised automated assessments model (PAAS), a trust-based assessment service that helps compute group assessments from the perspective of a specific community member. This computation essentially aggregates peer assessments, giving more weight to those peers that are trusted by the specific community member whom the automated assessments are computed for. How much this specific member trusts a peer is then based on the similarity or evaluation rate between his (past) assessments and the peer's (past) assessments over the same assignments.

The proposed work is an extension of the work carried out in [Gutierrez *et al.*, submitted for publication]. In fact, the COMAS model is a much more simplified model of the non-ordered case. It is much more simplified as it assumes that the probability of the similarity between two assessors is 1 for the aggregation of the similarities of past evaluations over the same objects. PAAS' use of probability distribution makes it a richer and more informative model as much more information is preserved in the calculations. Furthermore, PAAS computes the uncertainty of the automated assessments, helping suggesting which objects should be evaluated next in order to decrease the overall uncertainty of PAAS' calculations.

In COMAS, experimental results were conducted on a real classroom datasets as well as simulated data that considers different social network topologies (where we say students

assess some assignments of socially connected students). Results show that the COMAS method 1) is sound, i.e. the error of the suggested assessments decreases for increasing numbers of tutor assessments; and 2) scales for large numbers of students.

Future work on PAAS should follow a similar approach for evaluation, where the same real classroom datasets can be used as the groundtruth of marks, and we can then compare PAAS' automated assessments to that groundtruth.

Additionally, we could also test the ranking of marks (Section 2.5) by running experiments in a real classroom where we ask the tutor to evaluate assignments once in a random order and another time following the suggested ranking. This could help us check whether the error decreases faster in the latter case. Also, we expect to find that for a given acceptable uncertainty threshold, the tutor should evaluate less assignments in order to reach that threshold than evaluating randomly.

Acknowledgments

This work is supported by the CollectiveMind project (funded by the Spanish Ministry of Economy and Competitiveness, under grant number TEC2013-49430-EXP) and the PRAISE project (funded by the European Commission, under grant number 388770).

References

[de Alfaro and Shavlovsky, 2013] L. de Alfaro and M. Shavlovsky. Crowdgrader: Crowdsourcing the evaluation of homework assignments. *Thech. Report 1308.5273, arXiv.org*, 2013.

[Gutierrez *et al.*, submitted for publication] Patricia Gutierrez, Nardine Osman, and Carles Sierra. Trust-based community assessment. *Pattern Recognition Letters*, submitted for publication.

[Li *et al.*, 2003] Yuhua Li, Zuhair A. Bandar, and David McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. on Knowl. and Data Eng.*, 15(4):871–882, July 2003.

[Piech *et al.*, 2013] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned models of peer assessment in moocs. *Proc. of the 6th International Conference on Educational Data Mining (EDM 2013)*, 2013.

[Rubner *et al.*, 1998] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV 1998)*, ICCV '98, pages 59–, Washington, DC, USA, 1998. IEEE Computer Society.

[Sierra and Debenham, 2005] Carles Sierra and John Debenham. An information-based model for trust. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 497–504, New York, NY, USA, 2005. ACM.

[Upton and Cook, 2008] G. Upton and I. Cook. *A Dictionary of Statistics*. Oxford Paperback Reference. OUP Oxford, 2008.

[Walsh, 2014] Toby Walsh. The peerrank method for peer assessment. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 909–914. IOS Press, 2014.

[Wu *et al.*, 2015] J. Wu, F. Chiclana, and E. Herrera-Viedma. Trust based consensus model for social network in an incompletelinguistic information context. *Applied Soft Computing*, 2015.

Revealing and Interpreting Crowd Stories in Online Social Environments

Chris Kiefer, Matthew Yee-King and Mark d'Inverno
Department of Computing,
Goldsmiths, University of London
m.yee-king@gold.ac.uk

Abstract

The underlying patterns in large scale social media datasets can reveal valuable information for interaction designers and researchers, both as part of realtime interactive systems and for post-hoc analysis. *Music Circle* is a social media platform aimed at researching the role of community feedback in online learning environments. A large dataset was collected when the platform was used as part of a *Massive Open Online Course* (MOOC). We developed a novel analysis technique for observing global patterns in the behaviour of students. The technique employs network theory techniques to view student activity as an interconnected complex system, and observes the temporal dynamics of network metrics to create timelines which are clustered into groups using unsupervised learning methods. This approach highlighted global trends and groups of outliers that needed further attention or intervention.

1 Introduction

Online social activity has become a fundamental part of many interactive systems, either explicitly as part of their intended design, or implicitly as part of external and pervasive social media networks. With social activity comes large or massive scale data, which describes interactions between individuals mediated through a variety of possible formats. These datasets can reveal stories about individuals and groups that may be of high significance to stakeholders; for interaction designers, this data can show aspects of behaviour that reveal design problems, suggest design solutions and highlight directions for future iterations. For researchers, this data can give us a broad understanding of trends in behaviour within the context of specific technological environments. Large datasets also present significant challenges in analysis, with the scale of raw data often making direct human interpretation an intractable task. However, by bringing computational analysis into the loop, we can attempt to sculpt the raw data into new forms that, while not necessarily giving absolute answers, present data in a suitable format for further (human) interpretation and criticism.

Music Circle is an online social platform, aimed at exploring ways of understanding and enhancing learning through community feedback. For six weeks in the summer of 2014 it was used to support a Coursera MOOC, *'Creative Programming for Digital Media and Mobile Apps'*[1]. A substantial and detailed log of student interactions was collected. While there were specific questions that could be asked of the data, such a large and complex set of interactions could most likely hold some interesting and unexpected results, and it seemed pertinent to follow a bottom-up approach to data analysis, by letting patterns emerge rather than imposing them. To this purpose, a set of techniques was developed that attempted to elucidate the broad patterns and temporal dynamics of crowd behaviour that occurred during the period of the MOOC, to transform the raw data into a format that would give the research and design teams a deeper understanding of student interactions within the Music Circle environment.

A novel approach was developed, which leveraged network analysis and machine learning techniques to cluster temporal data. We outline the development of this technique and present and critique the results. The following sections address the research questions that were encountered during this development process: how can social media data be encoded into a human readable form that describes temporal patterns in actor behaviour? How can network analysis techniques enhance this encoding? What are good ways to present this data for interpretation by stakeholders?

We present this research as a technique for eliciting information from large datasets for analysis by stakeholders and domain experts, rather than as a process which will supply absolute answers concerning student behaviour. In this light, we do not attempt to provide a quantitative evaluation of the effectiveness of the findings, but try to show, through cross-checking of results in a post-hoc analysis of the Music Circle MOOC data, the potential strengths of our method for use in future projects.

2 Related Work

Our approach is rooted in a network theory perspective. Jiawei et. al [Han *et al.*, 2012] review data mining in this context, proposing that we can extract much more valuable

[1] https://www.coursera.org/course/digitalmedia

information from a database by viewing it as a heterogeneous information network rather than a homogenous data repository. We draw on techniques highlighted by Holme and Saramäki [Holme and Saramäki, 2012]; they review the emerging field of temporal networks, looking at techniques for analysing how network topology changes over time, and how temporal information flows. There have been varied approaches to social network analysis, for example, Gottron and Pickhardt[Gottron *et al.*, 2013] explore techniques for temporal analysis of social data, Gilbert et. al. use statistical methods to analyse Pinterest [Gilbert *et al.*, 2013], and Diya et. al. [Yang *et al.*, 2013] look for causes of student dropouts in MOOCs using a network theory approach. Rowe et. al. [Rowe *et al.*, 2013] outline their technique for modelling and analysing the behaviour of users in online communities. They focus on defining individual role categories, and look at how the global composition of these roles changes over time. Chao et. al [Chau *et al.*, 2011] look at the intersection between HCI and data mining, investigating interactive machine learning approaches, and sensemaking.

3 Music Circle

Our system (pictured in figure 1) allows students to share and discuss creative work, and acts as a research platform for studying the role of social media in learning [Brenton *et al.*, 2014]. The key feature is the *Social Timeline*, an online environment for annotating and discussing time-based media. The Social Timeline allows students to highlight and comment on sections of time-based media, and to further discuss these comments. The website has been used in a variety of scenarios to explore creative feedback [d'Inverno and Still, 2014] between students, including jazz piano tuition and as a rehearsal support tool for musical ensembles.

Over the course of a MOOC in the summer of 2014, the website was employed for students to discuss, share feedback on and peer assess videos of their coursework pieces. During the six week course, the students were required to submit a piece of coursework every two weeks. Each coursework brief asked them to program a software application, and to submit a video demo of their application to the Music Circle website for peer assessment. Students were also asked to peer assess three other peoples' work for each submission. As part of the peer assessment process, they could discuss other students' work through the website.

To give an overview of the course statistics, during a six week period, 3716 users registered with the website. Of these, 3558 viewed one or more videos, 827 made one or more comments, and 258 made one or more replies to comments. 2898 videos were submitted for three separate assessments, and were viewed a total of 112,189 times. 7370 comments were made, along with 978 replies. Detailed log data was collected, including timestamped records of all discussions and all media viewing activity.

4 Encoding Stories

Having collected the data, we needed to present it in a format that was both interpretable by humans, and cluster-able by a computer. We built networks of data relating to singular concepts, and observed several network metrics as they evolved over time. In this way, we could use network measurements that put an individual's actions into a global interconnected context, rather than observing them in local scope.

Two sets of networks were built, that separately represented commenting and viewing activity. Each set consisted of networks that evolved in two hour windows over the period of the MOOC, giving 503 networks in each set. This two hour period was chosen as a compromise between time resolution and practical limitations in data processing capacity. The networks had directed and weighted connections; each node represented a student, with weighted links representing the numbers of comments or views made from one student to another.

In this analysis, a set of four metrics were chosen for observation from each network. The first two were simple metrics which sum the (a) incoming and (b) outgoing weights of each node. These could also be calculated without using a network. The next was (c) betweenness centrality. This metric was chosen as it provides interesting representations of each user's importance within the global context of the network, based on how much information flows through their node. It shows the extent that the actor is positioned on the shortest path between other pairs of nodes in the network [Leydesdorff, 2007]. Betweenness centrality is calculated based on link direction and weight, thereby using the full information available in the networks we constructed. The last metric was a calculation of each node's (d) *HITS authority*. This was calculated with the HITS algorithm [Kleinberg *et al.*, 1999], which gives a measurement of the importance of a node based on link structure. More specifically the algorithm gives each node two co-dependent scores; a *hub* score based on the authority of nodes that link to it, and an *authority* score based on how the degree to which the nodes that point to it are hubs.

These four metrics were observed for each two hour iteration of the networks, giving each student a set of timelines, one for each metric. The analysis provided a rich data set for further exploration. Network analysis was carried out using *iPython* with the *NetworkX* library.

5 Discovering Themes

Having collected the sets of timelines for each user, the next step was to cluster these timelines into similar groups to reveal underlying patterns. An exploratory approach was taken, searching for interesting features in the data set by creating both clusters of single features and clusters of compound features in order to reveal correlations between groups of metrics. Useful clustering results were obtained using two methods: k-means alone, and k-means with unsupervised pretraining using Restricted Boltzmann Machines. In the latter case, RBMs were used to find sparse, low dimensional representations of the salient features in the data, before k-means clustered these features. We used the Extended RBM from the Oger toolbox [Verstraeten *et al.*, 2012], with gaussian visible units for our continuous valued data. The RBMs were trained with guidance from [Hinton, 2010].

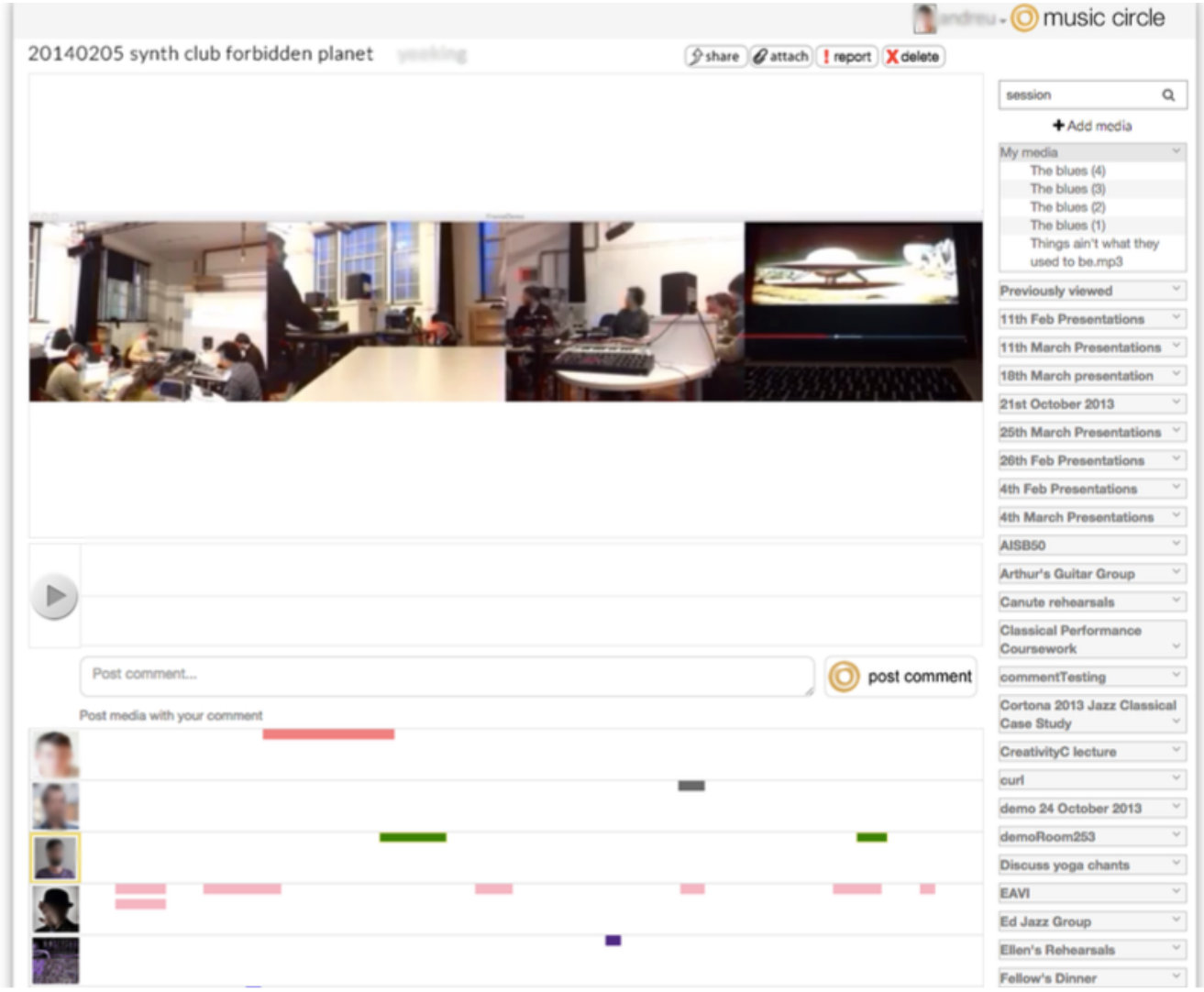


Figure 1: A Screen Shot of the Music Circle Website

6 Visualising Crowd Behaviour

Having calculated clusters, we visualised them in two ways. Simple graphs of means and variances for each cluster group (e.g. figure 2) gave an easily interpretable summary. A more complex view showing members of individual clusters was presented in sets of polar graphs, where each graph displayed every individual timeline in a cluster, superimposed with semi-transparent colouring to show patterns of density (e.g. figure 3). Upon identifying a cluster of interest, a set of graphs was generated to show the cluster mean for each metric, compared to the global means (e.g. figure 4).

7 Results

An exploratory approach was taken to analysing the data, visualising features separately and in combination to find clusters of possible interest. The following examples describe salient outcomes of this process.

7.1 Example A: Betweenness Centrality

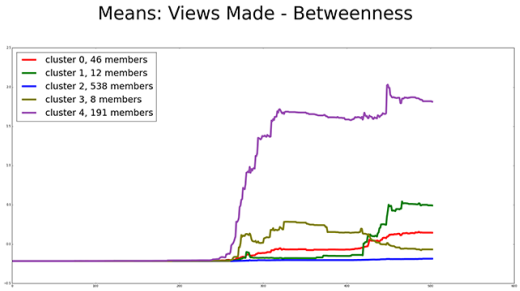


Figure 2: Mean View Betweenness Timelines, in 5 Clusters

Figure 2 shows the means of 5 clusters of betweenness centrality timelines from the *views* network, generated by training an RBM with 503 visible units and 5 hidden units, the output of which was clustered with k-means. In this context, we could consider betweenness centrality to indicate the extent to which a student is engaged in a community of other students who are active in viewing each others' work. A large group of low activity users is shown in cluster 2, which is what would be expected in a social network dataset. A smaller group of high activity users is highlighted in cluster 4 (shown in more detail in figure 3). This timing of this higher viewing activity correlates with an incentive being offered to students for engaging in forum activity. Cluster 3's value is dropping while other clusters are rising, indicating that this cluster may include students who need attention in some way. Further analysis shows that the number of comments received by this group is below the global average (see figure 4), strengthening the case that this group may need some sort of help or motivation.

7.2 Example B: HITS Authority

Clusters of the HITS authority timelines for the last 40% of the course were clustered using k-means (shown in figure 5). The graph shows that the students in cluster 4 have a steadily

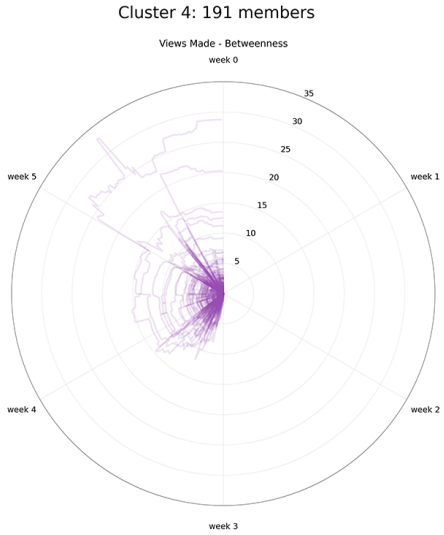


Figure 3: A polar plot of superimposed timelines from a single cluster

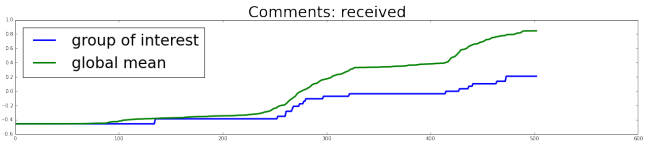


Figure 4: The cluster mean vs global mean for 'comments made'

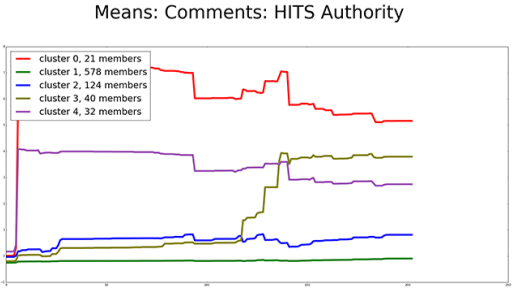


Figure 5: Means of HITS authority for the comments network

declining authority, which may indicate the students in this group have low activity and are therefore becoming less important community members. The concern is verified when looking at their viewing timelines; they are significantly below the global mean for betweenness centrality of views.

7.3 Example C: A Compound Feature

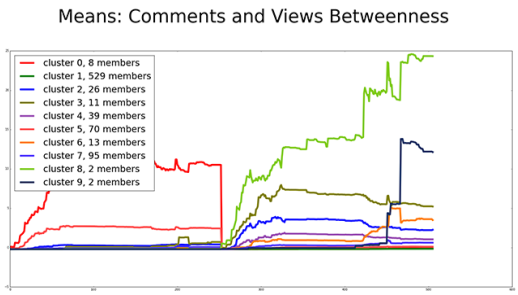


Figure 6:

Figure 6 shows 10 clusters of a compound feature, made with k-means. The first half is a betweenness centrality from the views network, and the second half is the same metric from the comments network. By joining features in this way, the clustering process can pick out potentially interesting correlations or contrasts between the sources. In this example, we can see that the students in cluster 5 have a relatively high value for views but a very low value for comments. This could indicate students who are potentially active, but are part of low activity peer communities, and would perhaps benefit from being introduced to new peers.

8 Discussion

By exploring and clustering the sets of timelines describing behaviour on Music Circle, it was possible to reveal global patterns of crowd behaviour in the specific context of the network metrics we observed. The clusters also highlighted groups of outliers; understanding these small groups can be of great value in trying to understand a complex social system, both in terms of isolating problems, and understanding positive deviance [Ramalingam, 2013]. The technique can be used both post-hoc and for live analysis. Post-hoc analysis can help us to understand crowd behaviour in order to improve the design of future iterations. Live analysis has a number of possible uses in the context of our platform; outlying groups may predict students who are likely to drop out or disengage, and need some contact, reward or support from peers and teachers. Outliers may also highlight successful groups who we may try to link with new peers to strengthen the overall community of learning. A further use is to give students live feedback of their status in terms of these metrics, in order to aid their learning or motivate them. For example, a live timeline of centrality in the comments network, together with a summary based on cluster membership, could provide a good motivator to increase commenting activities.

The first two examples in particular demonstrate the potential strengths of our analysis technique. Both highlight groups of interest, which are validated by patterns in other metrics on other domains. e.g. in example A, the betweenness centrality clusters for viewing behaviour highlight a group that may need attention. Further investigation of the commenting network reveals that this group is less active at making comments, compared to the global mean. Examples A and B also demonstrate the values of analysing our data from a network perspective, viewing user activity as a complex evolving system of interconnected nodes. In example A, the clusters highlight a group whose betweenness centrality is dropping progressively. Observing non-network based metrics for this group, i.e. the number of views made and received, the timelines for this group do not differ greatly from the global average, so this group would not show up in any clusters. However, the highlighting of the group is validated by their inactivity in commenting. In example B, a group is revealed whose HITS authority for comments is dropping. Again, this would be difficult to spot from this group's number of comments made and received, which are close to the global average, but the choice is validated by revealing their low viewing activity. Overall, network analysis algorithms such as betweenness centrality and HITS evaluate each node in the wider context of a complex system. This means these metrics are much more sensitive to global events in the network, and can reveal dynamics that locally scoped measurements may fail to. The results show the merits of temporal analysis of these network metrics; the clusters of interest were highlighted by discovering anomalies in temporal dynamics, and reveal more detailed information compared to instantaneous analysis.

A challenge of using this system is in interpretation. To fully interpret a graph of clusters, it is necessary to understand the network analysis metric being presented, along with its meaning in the context of the network and in the wider context of the source domain of the data. For example, to understand betweenness centrality of commenting activity, we need to understand the concept of this measurement along with the network theory that supports it, and we also need to understand how people are connected by comments on Music Circle and the affordances of the interface that allow activity to propagate though the network of students. It's also a challenge to present the clusters in an optimal way. The means and variance give a good idea of general trends but miss some details. The graphs of superimposed timelines can become dense and difficult to compare, but do give much more detail. Conducting analysis with both of these perspectives seems a good compromise, but ideally an interactive tool would be very useful.

9 Conclusions

The motivation for this project was to reveal patterns of global crowd behaviour based on a large scale database of social and educational activity. Our approach was to look at simple information through the perspective of network analysis. We observed how a variety of network analysis measurements vary over time, and then undertook an exploratory analysis of these timelines though clustering. The clusters highlighted interesting global behaviours of groups of users, and also re-

vealed smaller groups of outliers that may need some sort of intervention or attention. The strength of this technique is demonstrated in examples where the highlighted clusters were shown to need attention though cross checking with other data sources. The possibilities of our technique were demonstrated through post-hoc analysis of forum data. The next step would be to apply this technique on a live forum, and observe the effects of any pedagogical interventions that are made based on the analysis of the resulting data.

10 Future Work

The analysis highlights two areas which could benefit from further development. Firstly, the development of presentation tools to aid human analysis of the clusters. Secondly, the network analysis algorithms employed here have been successful in highlighting clusters but also add an extra layer of interpretation. It would be interesting to investigate the development of domain specific networks measurements, whose output is closely matched to the semantics of the forum behaviour being observed.

Acknowledgments

The work reported in this paper is part of the PRAISE (Practice and Performance Analysis Inspiring Social Education) project which is funded under the EU FP7 Technology Enhanced Learning programme, grant agreement number 318770.

References

[Brenton *et al.*, 2014] Harry Brenton, Matthew Yee-King, Andreu Grimalt-Reynes, Marco Gilles, Maria Krivenski, and Mark d'Inverno. A social timeline for exchanging feedback about musical performances. In *Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI 2014-Sand, Sea and Sky-Holiday HCI*, pages 281–286. BCS, 2014.

[Chau *et al.*, 2011] Duen Horng Chau, Aniket Kittur, Jason I Hong, and Christos Faloutsos. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 167–176. ACM, 2011.

[d'Inverno and Still, 2014] Mark d'Inverno and Arthur Still. Creative feedback: a manifesto for social learning. In *Proceedings of the Workshops held at Educational Data Mining 2014 conference*, 2014.

[Gilbert *et al.*, 2013] Eric Gilbert, Saeideh Bakhshi, Shuo Chang, and Loren Terveen. "i need to try this"?: A statistical overview of pinterest. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2427–2436, New York, NY, USA, 2013. ACM.

[Gottron *et al.*, 2013] Thomas Gottron, Olaf Radcke, and Rene Pickhardt. On the temporal dynamics of influence on the social semantic web. In *Semantic Web and Web Science*, pages 75–87. Springer, 2013.

[Han *et al.*, 2012] Jiawei Han, Yizhou Sun, Xifeng Yan, and Philip S Yu. Mining knowledge from data: An information network analysis approach. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1214–1217. IEEE, 2012.

[Hinton, 2010] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. Technical Report 1, University of Toronto, August 2010.

[Holme and Saramäki, 2012] Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.

[Kleinberg *et al.*, 1999] Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. The web as a graph: measurements, models, and methods. In *Computing and combinatorics*, pages 1–17. Springer, 1999.

[Leydesdorff, 2007] Loet Leydesdorff. Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. *Journal of the American Society for Information Science and Technology*, 58(9):1303–1319, 2007.

[Ramalingam, 2013] Ben Ramalingam. *Aid on the edge of chaos: rethinking international cooperation in a complex world*. Oxford University Press, 2013.

[Rowe *et al.*, 2013] Mattew Rowe, Miriam Fernandez, and Harith Alani. Modelling and analysis of user behaviour in online communities: Ieee computer society special technical community on social networking e-letter. *IEEE Computer Society Special Technical Community on Social Networking E-Letter*, 1(2), May 2013.

[Verstraeten *et al.*, 2012] David Verstraeten, Benjamin Schrauwen, Sander Dieleman, Philemon Brakel, Pieter Buteneers, and Dejan Pecevski. Oger: modular learning architectures for large-scale sequential processing. *The Journal of Machine Learning Research*, 13(1):2995–2998, 2012.

[Yang *et al.*, 2013] Diyi Yang, Tanmay Sinha, David Adamson, and Carolyn Penstein Rosé. Turn on, tune in, drop out: Anticipating student dropouts in massive open online courses. In *Proceedings of the 2013 NIPS Data-Driven Education Workshop*, 2013.

Author Index