Surprise Walks: Encouraging users towards novel concepts with sequential suggestions

Kazjon Grace[1], Mary Lou Maher[2], Nicholas Davis[2], and Omar Eltayeby[2]
[1]The University of Sydney, and [2]UNC Charlotte
[1]Australia, and [2]USA
kazjon.grace@sydney.edu.au, {m.maher, ndavis64, oeltayeb}@uncc.edu

Abstract

In this paper we present a proof-of-concept of how co-creative systems could guide their users to appreciate artefacts that are currently too novel. Given that too-novel artefacts are off-putting, and domain experience reduces novelty, this situation will arise often when a co-creative system has more domain experience than its user. We present some experiments demonstrating a strategy for generating sequences of concepts to present to users. These sequences are designed to provide the necessary background to allow users to appreciate a highly-novel "target" artefact. Our strategy is based on generating and then traversing "surprise space", a form of conceptual space in which concepts which are surprising in the same contexts are proximal. We implement this strategy, which we call a "surprise walk", in the domain of recipes using a word embedding algorithm with a modified objective function that co-locates features that are similarly surprising.

Introduction

Consider the case where a human and computer are collaborating on a creative task (aka "co-creativity"), but the latter knows more than the former. Where we are today, at the very beginning of usable co-creative systems, that might seem like an edge case. We contend, however, that in time it might describe the majority of such interactions. Imagine a future in which co-creative systems are commonplace: it is likely that the majority of their users will not be experts. It follows that co-creative systems will often possess knowledge their users do not, even discounting situations in which they are explicitly being used for education.

This creates a challenge for systems that generate content more novel than their users are currently prepared to accept. Under the Wundt curve model (Berlyne 1966; Saunders and Gero 2001), there is a peak level of novelty at which positive affective response is maximised. Either side of that peak the response becomes negative: either too boring (insufficient novelty) or too alien (overwhelming novelty). Creative systems operating with more knowledge than their users will often generate artefacts that are desirably novel to the system, but (if we accept the Wundt curve model) overwhelmingly so to their users. Greater knowledge would lead to more accurate expectations, and thus less surprise.

If the human has decision-making power in the creative task, as is common in co-creative systems, then a co-creative system must convince its users of the benefits of its creations. How could a co-creative system "guide" a human towards a creative (i.e. novel and valuable, (Newell, Shaw, and Simon 1959)) region of the space of possible artefacts, even if those artefacts were currently overwhelmingly novel to the user? One answer is for systems to seek "user appropriate" rather than maximal novelty. Another is persuasion.

In this paper we explore how computationally creative systems might persuade humans to appreciate more novel artefacts. We propose "surprise walks", a strategy for generating sequences of increasingly surprising concepts. These sequences start with a goal concept that the system desires the user be able to appreciate. The strategy is then to work backwards, decreasing the level of surprise, until a concept that the user can appreciate is reached. A creative system could then expose its user to artefacts exhibiting each concept in turn. Where necessary, multiple artefacts exhibiting a concept could be presented until the user appears to have comprehended or accepted it. The intent is to pique user curiosity over time, and maintain that curiosity state while working towards a goal (Grace and Maher 2015). That the user is being taken for a "surprise walk" may or may not be communicated to the user, which raises a variety of ethical issues which we return to in the discussion.

We present a model of the surprise walk process and additionally introduce the concept of "surprise space" on which the process is based. A "surprise space" is a specialised kind of conceptual space in which proximal concepts are *similarly surprising*, rather than being literally similar themselves. We also present a prototype implementation of a surprise walk generator, capable of accepting a target surprise and a simple artificial user profile and outputting sequences of concepts. We present and discuss the results of this prototype, comparing the sequences that can be generated using a surprise space to those generated by the same process using a conceptual space based on literal similarity.

Background

This research occurs at the intersection of two literatures: co-creativity and computational models of surprise and curiosity. To date, most research in co-creative systems has not explicitly considered the idea of imbuing such systems with the desire to spark curiosity. Similarly, most research in computational curiosity has not considered the context of co-creativity.

Co-creative systems

A variety of co-creative systems are able to influence their user's behaviour. The Drawing Apprentice (DA) is a co-creative drawing partner that collaborates with users on a shared drawing (Davis et al. 2016; 2018). The system analyses the user's input and responds with complementary objects to inspire the user's creativity and sustain engagement over time. The Sentient Sketchbook is a co-creative game level design tool that leverages user input to generate design alternatives that may surprise the user and support their creativity (Liapis, Yannakakis, and Togelius 2013). Clark et al. (2018) describe a machine-in-the-loop writing system that provides surprising and unpredictable output designed to inspire user creativity. Similar systems include Creative Help (Roemmele and Gordon 2015) and Say Anything (Swanson and Gordon 2012).

In none of the above systems is there a capacity to reason beyond the next step in designing the current artefact. That is not an criticism, doing so is simply out of their scope. They assist human creators by providing in-the-moment suggestions. This research explores a way for co-creative systems to form longer-term goals.

Computational surprise

The concepts of novelty, unexpectedness and surprise have been the subject of many definitions in the computational creativity, artificial intelligence and cognitive science literatures. For the purposes of this study we define novelty as the degree to which an artefact differs from those that have come before within that creative domain. There are many ways to operationalise that definition, but building on our previous work we argue that the best way to do so is by quantifying the expectations of the agents acting in that domain, and then measuring the degree to which those expectations are violated by an artefact (Grace and Maher 2014). We call this an *unexpectedness* based approach to novelty. Similar approaches have been adopted by Macedo and Cardoso 2001 and Gravina et al 2016.

In most of our work expectations are defined in terms of sets of features that co-occur regularly, with unexpected artefacts being those which exhibit sets of features that co-occur only infrequently. "Surprise" is an agent's response to unexpectedness, although in most contexts this can be used interchangeably with unexpectedness. We measure the amount of surprise using the negative base-2 log of ratio of the co-occurrence probability of those features to their probability of them occurring separately. "Surprise walks" are thus an exploration of how a co-creative system could expose an individual to a sequence of surprising artefacts, each not only similar but more unlikely than the last.

Computational curiosity

Berlyne (1966) describes the prevailing psychological theories of curiosity as curiosity-as-state and curiosity-as-trait. Curiosity-as-trait refers to an innate ability of a person, and individuals differ in how much curiosity they have. Curiosity-as-state refers to a motivational state of a person that causes the person to seek novel stimuli, and it varies within each person according to their context. Curiosity-as-state is malleable: curiosity can be encouraged by external events or contexts. A computational model of state curiosity is one that seeks surprising events or objects and in co-creativity a computational model of surprise can present stimuli that encourages user curiosity. Curiosity-as-state has been integrated into cognitive systems in the past, such as Saunders and Gero (2001) and Merrick and Maher (2009) .

Berlyne additionally proposed that state curiosity can be considered along two dimensions: epistemic vs perceptual, and diversive vs specific. In the first dimension, perceptual curiosity is the drive towards novel sensory stimuli and epistemic curiosity is the drive to acquire new knowledge. Our surprise walks could theoretically be applied in either case, but we are exploring epistemic applications. Along the second dimension, Berlyne describes diversive curiosity as unguided search for any new information or stimuli and specific curiosity is search for a novel solution to a specific problem or goal. The majority of current models of computational curiosity are diversive in nature, such as Saunders and Gero and Merrick and Maher mentioned above as well as Schmidhuber (2010). Our surprise walks are adopting the concept of specific curiosity: how a system could influence a user towards a novel goal.

Surprise Spaces

A conceptual space in a creative domain captures the ordering principles or underlying structure of that domain's concepts. In some conceptual spaces the artefacts are described in terms of dimensions that are meaningful to that domain (Gärdenfors 2004). If a conceptual space is constructed in this manner then the "concepts" are dimensions, and each point represents a hypothetical artefact. In other approaches globally meaningful dimensions are not required, instead proximal concepts are always similar (Boden 1996). In this approach each point in space represents a concept, and an artefact is composed from one or more concepts.

A surprise space is a particular kind of conceptual space in the latter tradition: proximity implies similarity. However the concepts that are distributed through that space are combinations of artefact features, each of which is assigned a surprise based on measures described in our previous work (Grace et al. 2017). As a simple example, consider a surprise space in the domain of recipes in which each point represents the combination of two ingredients. Some of those combinations (such as onion and garlic) will be of low surprise, while others (such as chocolate and garlic) will not. A surprise space need not be constructed of these simple unordered pairwise combinations of features, but could instead contain any combination of two or more elements that is meaningful to the domain: consecutive phrases of music, visual features combined with a particular caption word, or triplets of named entities appearing together in news articles. A surprise space is intended to augment, rather than replace any other form of conceptual space in a creative system's reasoning. We do not suggest that this way of constructing conceptual spaces is in any way superior to any other – it is simply different.

The principle of organisation in surprise space is the similarity between surprises. By carefully traversing this space we could construct a sequence of surprises that are increasingly but also similarly surprising. That sequence could transport a user from the borders of their current knowledge to some as-yet-too-alien combination. This journey through surprise space, which we call a "surprise walk", leverages the unique structure of a surprise space as a metacognitive aid. It guides a creative system's behaviour as a means for influencing the behaviour of its human collaborators.

But what does it mean for two surprises to be similar? There are several possible approaches here, and we prototype two in the proof-of-concept detailed below. The simplest is to average the similarity of each feature (or set of features) in the combination, using the best possible mapping between features to do so. For example, assume that each point in surprise-space represents a combination of two or more ingredients in a recipe. Given (A,B) and (C,D) as two such combinations, we can take the similarity between the two surprises as being:

$$max(s(A,C) + s(B,D)), (s(A,D) + s(B,C)))/2$$

Where $s(x, y)$ as a similarity measure for features x and y. Note, again, that this is specifically the similarity between two *surprises*, (A, B) and (C, D).

An alternative approach would be to construct a similarity measure between surprises. This is akin to comparing between two differences: how similar is the difference between A and B to the difference between C and D, to take the example above? In our recipe example this could be measured using a physiological model of taste, a molecular gastronomical model of chemical compounds, or the co-occurrence of ingredients. We introduce a hypothetically domain-independent approach below that performs this kind of surprise-to-surprise comparison based on whether the ingredients are *surprising in similar contexts*. Let's say A is soy sauce, B is chocolate, C is mushrooms and D is icing sugar (confectioner's sugar in North American English). Is the way soy sauce differs from chocolate similar to the way mushrooms differ from icing sugar? Our prototype says yes: soy sauce is surprising when combined with a similar list of things as mushrooms are, and the same with chocolate and icing sugar. For example, both soy sauce and mushrooms are surprising in combination with vanilla, apples, and barbecue sauce. Similarly, chocolate and icing sugar are both surprising in combination with steak, black pepper, and tofu.

In our proof-of-concept implementation we have implemented both approaches: literal feature similarity comparison as well as comparing the similarity of surprises directly.

Surprise Walks: Navigating surprise spaces

Our motivation in conceiving of "surprise walks" is to explore how co-creative systems could encourage their users towards appreciating concepts that they could currently consider too novel. We define a surprise walk as a sequence of combinations in a surprise space that a) are of monotonically increasing surprise, b) are sequentially proximal in the space, c) start with a combination familiar to the user and d) end with a target combination of (currently) overwhelming novelty. That target combination is not only novel, but so novel that the user cannot or will not appreciate it: it is off the right shoulder of their personal Wundt curve.

Additional constraints on the sequences might be desirable, such as ensuring that adjacent elements are not too dissimilar in their surprise ratings. The intent is that these sequences act as a long-term plan for the behaviour of the co-creative system. They could allow it to curate the new experiences of their human user and thereby influence that user's Wundt curve until the target combination is no longer overwhelming. This definition is sufficiently broad to permit a large variety of approaches to sequence generation. We describe one such approach below in the domain of recipes.

s-GloVe: A prototype surprise space

Our prototype surprise walk system is based on a word embedding algorithm called "GloVe" (Pennington, Socher, and Manning 2014), used for representing each word in a corpus of documents as a vector of numbers. We call our surprise-based modification of it "s-GloVe". Word embedding algorithms map each word that occurs in a corpus of documents (typically one in which each document is represented as a bag of words, i.e. a count of all words that occur, ignoring word order) into an abstract continuous space. This space typically has a few hundred dimensions. We selected GloVe for this work as it approximates the matrix of co-occurrences between features, a desirable quality in a model of unexpectedness. We treat each ingredient as a "word".

Representing each word as a continuous vector allows for capturing similarity between words: similar words are proximal in vector-space. The most common approach to measuring the "similarity" between two words is based on the concept of *distributional semantics*, or the idea that you can "know a word by the company it keeps" (Firth 1957). More precisely, distributional semantics states that similar words have similar distributions over what other words are likely to occur nearby (Harris 1954): they occur in similar sentences. Constructing a word embedding model such that words with similar contexts occur nearby in vector space makes all kinds of similarity-based tasks easier, including clustering, thematic analysis, document classification, and augmenting the training of other machine learning algorithms.

GloVe has become a standard for word embeddings as it is simple, scalable and robust. It operates by learning a vector of arbitrary numbers for each word in the corpus. Its objective is to construct those vectors such that the vectors of any two words can be used (via a mathematical transformation) to calculate how those co-occur. What exactly it means for two words to "co-occur" is dataset specific: it could be that they are both within the same sentence in a news article, within the same line in a poem, or within the same section in a scientific paper. The result is that the vector representation for each word encodes how that word correlates with every other word. When these word vectors are interpreted as points in space, nearby words co-occur with all other words in similar ways. Or, to put it another way, they share distributional semantics. GloVe uses gradient descent to construct its vectors, with the objective of minimising the difference between the true co-occurrence of words and the one reconstructed from the word vectors.

A full description of the GloVe algorithm can be found in the original paper, but two points are relevant to how we modified the algorithm to discover similar surprises. Firstly, GloVe's vectors exhibit locally linear relationships between words that capture their meanings. This means that the differences between similar pairs of words are themselves similar. The difference between woman and man (the subtraction of those two vectors) is similar to the difference between queen and king, or between aunt and uncle. Similarly, the differences between US cities and their zip codes are all similar, as are the differences between Fortune 500 companies and their CEOs. This extends also to grammatical structure, with the difference between comparative and superlative forms of the same adjective (e.g. "softer" vs "softest") being highly similar. We exploit this property in our prototype.

Secondly, to speed up training and produce more robust vectors the original GloVe algorithm lowers the impact of reconstruction error for rare words using a weighting function. It is by replacing this weighting function that we re-imagine GloVe as a space of similar surprises.

s-Glove: The distributional semantics of surprise

GloVe captures the meaning of words by quantifying the company they keep. S-GloVe captures the way words are surprising by quantifying the company they don't, or rather, the company in which they are unexpected. This is still a kind of distributional semantics, as it defines words by the statistical properties of their context. In practice, however, it leverages almost the opposite information to the basic GloVe approach. S-GloVe encodes the co-occurrence between only those word-pairs that are surprising. In doing so it effectively disregards all the commonly co-occurring words, which are the key information leveraged in the majority of distributional models. This creates a space where nearby surprises are similar because of why they are surprising, not because of what they are. This could permit a system to reason about *why* a particular combination is surprising/novel.

The GloVe cost function includes a weighting against rare words (Pennington, Socher, and Manning 2014). For the technical details consult the original paper, but in short it reduces the impact of the error in reconstructing the co-occurrence between words if that co-occurrence is low. The effect of this is that rare co-occurrences are not encoded as strongly in the word vectors, and do not affect word-word similarity as much. We replace this with a function that reduces the impact of co-occurrences which are unsurprising.

We use a test for statistical significance – the one-tailed version of Fisher's exact test – to quantify the evidence for whether a pair of words occur less frequently together than one would expect were they independent. This test draws from the word occurrence and co-occurrence data, and provides a p-value for the chance that they are actually significantly surprising. A sufficiently low p-value lets us reject the null hypothesis that this pair of words is not surprising (i.e. either independent or actually more likely to occur together). The specific weighting function we use in s-GloVe, which replaces $f(X_{ij})$ in Pennington et al (2014), is:

$$f(X_{ij}) = 1 - min(p_{ij}, \alpha) \tag{1}$$

where $p_{i,j}$ is the p-value of the left tail of the Fisher test for independence of words i and j, and α is a parameter controlling how small the impact of unsurprising word-pairs will be on the word vectors. As α approaches 0 unsurprising word pairs have effectively no impact as p-values above 0.999 are common. We used $\alpha = 0.1$ in our tests after some experimentation, as with higher values the s-GloVe space began to more strongly resemble the original GloVe space.

Dataset Description & Preprocessing Pipeline

We began with the Now You're Cooking dataset[1], as used in Kiddon et al 2016. The dataset contains around 80,000 unique recipes that have been shared on the Internet since the 1990's. The recipes are provided with their names, ingredients, quantities, units, preparation steps and tags. In our experiments we use the ingredient set and cuisine tags only, discarding for now the titles, quantities and preparation steps. We treated each ingredient, post-processing, as a single feature in our model, such as "white wine" or "parmesan cheese".

We used the New York Times' ingredient-phrase tagger[2] to extract from strings like "1/3 cup freshly shredded lettuce" the name of the ingredient itself (here "lettuce"). Manual cleaning of about 10% of recipes was required after this step, presumably due to differences between the NYT tagger's training data and our dataset. We also combined a number of less common ingredients (e.g. varieties of soy sauce or orange liqueur) into single categories for the purpose of simplicity. After parsing, cleaning, duplicate elimination, and deleting those recipes with less than three ingredients we ended up with 73,000 recipes. Figure 1 presents an example of our pre-processing, transforming complex ingredient strings into simple, corpus-coherent ingredient features.

Results: ingredient-ingredient similarity

To validate our ideas about what proximity in s-GloVe space represents, we compare the most similar ingredients to a target ingredient, i.e. the nearest neighbours in the vector space. We use cosine similarity in each case, and compare the same six ingredients between GloVe and s-Glove. In both cases (and throughout this paper) the ingredient vectors have 64 dimensions. In the case of the original GloVe paper the $xmax$ parameter is set to 0 to prevent de-emphasising rare words. We arbitrarily selected five highly dissimilar words as test cases: pine nuts (occurs in 485 recipes), cucumbers (1007 recipes), cayenne powder (1714 recipes),

[1] https://github.com/uwnlp/neural-checklist

[2] https://github.com/NYTimes/ingredient-phrase-tagger, as discussed at https://open.blogs.nytimes.com/2015/04/09/extracting-structured-data-from-recipes-using-conditional-random-fields

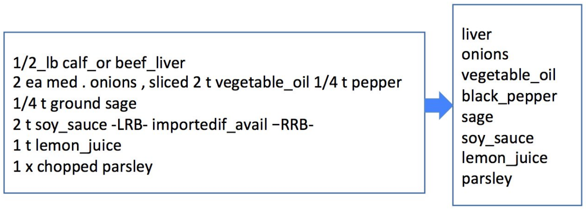

Figure 1: A set of "raw" recipe ingredients and the cleaned list used by our system, based on already-partially-preprocessed data by Kiddon et al (2016).

lentils (400 recipes), and apples (2270 recipes). The results are shown in Tables 1 through 5.

At the system's current level of development it is not yet feasible to objectively compare, via quantitative metrics or user feedback, the results of the s-GloVe surprise works to those of GloVe. Some interpretation must be permitted to judge the relative strengths and potential of the approaches. The results presented here are thus for the reader's own subjective digestion, although we believe they represent sufficient promise to continue investigating.

Table 1: Most similar to *"pine nuts"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
feta	0.60	tomatoes	0.63
olive oil	0.64	capers	0.67
currants	0.65	hazelnuts	0.69
zucchini	0.68	sesame seeds	0.69
basil	0.69	chili peppers	0.70

Table 2: Most similar to *"cucumbers"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
scallions	0.53	basil	0.60
radishes	0.53	peas	0.64
red onions	0.58	green beans	0.68
lettuce	0.61	beef broth	0.68
white vinegar	0.64	balsamic vinegar	0.71

These results show that the GloVe algorithm is capturing, as expected, the similarity between words that occur in similar contexts. Note that this is not the same as saying that they occur in the same recipes: lentils and brown rice may not occur together often, but when they occur separately they do so in the company of the same sorts of ingredients.

The s-Glove algorithm, however, is placing ingredients near to others that *are surprising when combined with the same sorts of ingredients*. GloVe suggests cucumbers are similar to radishes and red onions because (at least in our database) they occur in simmilar contexts, such as a variety of salads and Mediterranean dishes. S-Glove, however, finds cucumbers to be similar to ingredients like basil and peas, because it finds pairings like (cucumber, cocoa powder) and (cucumber, vanilla) to be highly similar to pairings

Table 3: Most similar to *"cayenne powder"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
cumin	0.47	lemon juice	0.69
jalapenos	0.47	lemons	0.70
paprika	0.49	lime juice	0.74
chili powder	0.52	salt	0.74
garlic	0.55	celery	0.75

Table 4: Most similar to *"lentils"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
brown rice	0.47	barbecue sauce	0.69
eggplant	0.47	steak	0.70
peas	0.49	brisket	0.74
kidney beans	0.52	ghee	0.74
barley	0.55	whiskey	0.75

like (peas, cocoa powder) and (peas, vanilla).

While this is only a cursory validation, we can conclude from this that the s-GloVe algorithm is able to measure the similarity between when ingredients are found surprising. We hypothesise, and in the following section explore, that this property can be used to generate interesting suggestions for guiding users towards more novel content.

Results: surprise-surprise similarity

We used the s-GloVe vector model described in the previous section and calculated the pairwise vector subtraction between all pairs of ingredients. This represented every combination of two ingredients, even those that had not occurred in any of our recipes, as a 64-dimensional vector. In this we are inspired by the linear local substructures observed in other word embedding experiments (Agres et al. 2015; McGregor, Purver, and Wiggins 2016).

This space satisfies our notion of a surprise space defined earlier. It is a space of combinations of concepts, each with a location and a surprise rating, in which proximity implies similarity between *why those combinations are surprising*. To give an example, the closest concept to the surprising combination of mozzarella and brown sugar (excluding those that share either) is sausage and molasses. Despite their similar locations the two combinations have quite different surprise values: mozzarella and brown sugar is quite surprising (surprise ≈5), while sausage and molasses is only slightly surprising (surprise ≈2).

As an initial exploration of the potential of this space, we

Table 5: Most similar to *"apples"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
raisins	0.46	ginger	0.68
cinnamon	0.53	icing sugar	0.71
nutmeg	0.54	walnuts	0.71
cranberries	0.55	currants	0.72
apple juice	0.57	cream	0.75

have implemented a simple – even trivial – surprise walk algorithm. Our motivation with surprise walks is to generate a sequence of combinations that can be incorporated into artefacts shown to a user. This sequence is intended to (gradually, perhaps with repeated exposure to artefacts containing each combination) guide the user towards being able to appreciate the "target" combination at the end of the sequence. That "target" is assumed to be outside of the "Wundt window" (i.e. off the right shoulder of the Wundt curve) for that user. It, along with a model of the user's familiarity with concepts in the domain, is the input to our model of surprise.

In our prototype we adopt a trivial synthetic user model: our prototype user is familiar with all surprises of less than 4 wows, as calculated by the method in (Grace and Maher 2016). This is based on the same co-occurrence matrix that is the input to the GloVe and s-Glove algorithms. Examples of combinations near this threshold are baking soda and tomatoes, apples and cumin, and lemongrass and walnuts. This threshold was chosen as it represents unusual but not (to the authors, at least) unheard of combinations, making it a good placeholder for the knowledge of a competent cook.

Our "surprise walk" algorithm, given a target surprise, first generates a list of the 25 nearest combinations. Those which are more surprising than the target are discarded. The system then iteratively greedily selects from that list the ingredient combination that most greatly reduces the surprise of the target without reducing it by more than a pre-defined "maximum surprise difference. In our experiments we set this threshold to 3 wows. If the selected combination is not familiar to the user then it becomes the new target and the greedy selection repeats. That means that if a target surprise is rated at 9 wows, then the system will pick the least surprising combination from the list of nearby combinations that is at least 6 wows, then repeat the process with a threshold of 3 less than that. At this point the combination would likely be less than our 4-wow threshold for the dummy user, and the sequence generation process would terminate.

This search is both greedy and naive. It is undirected, and would likely not work well with a more complex user model. A more nuanced approach would be to use a heuristic search algorithm like A* to find a path between the target and the user's "familiarity boundary". Despite its simplicity, this approach lets us explore the potential of traversing surprise spaces to generate goals for co-creative systems. Goal (re-)formulation has been suggested as a critical capacity for creative systems related to both autonomy (Jennings 2010) and metacreativity (Linkola et al. 2017).

Table 6 shows the result of a simple surprise walk on an ingredient combination that is only moderately surprising: bananas and basil. Both GloVe and s-Glove suggest one single combination as a sufficient stepping stone for the target combination. The suggested combination is familiar to our user (recall that our dummy user profile is familiar with all combinations of less than 4 wows) in both cases. This familiarity means that a co-creative system would likely only need to prompt the user with a few recipes before they are sufficiently primed as to appreciate bananas and basil.

GloVe suggested prompting the user with a combination of strawberries and thyme, highly literally similar to the target combination, but less surprising. Recipes involving this combination are typically pastries, jams[3], or cocktails. S-Glove suggested the less immediately obviously connected combination of applesauce and marjoram. Recipes involving this combination typically also involve pork, sausages, or game such as deer or partridge. s-GloVe considers applesauce and bananas to be quite similar (in terms of what they are surprising with), while GloVe does not. From this example it's hard to judge the quality of the two methods, although the difference in their approaches is clear.

Table 6: Surprise walks for *bananas and basil*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
strawberries	thyme	0.34	2.51
bananas	*basil*	–	*4.69*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
applesauce	marjoram	0.42	3.01
bananas	*basil*	–	*4.69*

A similar case seems to be occurring in Table 7, which shows the recommended steps for a user to appreciate the highly surprising combination of parmesan and vanilla. This combination is found in a few unusual salads and cakes as well as one weird pasta recipe. GloVe suggests the user approach it by first trying artichokes and icing sugar, then capsicum and icing sugar[4]. As in the first example these ingredients occur in the same contexts as those in the target.

Table 7: Surprise walks for *parmesan and vanilla*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
artichokes	icing sugar	0.29	2.23
capsicum	icing sugar	0.26	4.36
parmesan	*vanilla*	–	*7.14*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
mozzarella	figs	0.43	2.11
cheese	chocolate	0.39	4.25
parmesan	*vanilla*	–	*7.14*

By contrast s-Glove suggests that the user first try mozzarella and figs, then cheese and chocolate, then the target of parmesan and vanilla. Note that "cheese" here seems, on manual inspection of the dataset, to refer to the mild cheddar that is the typical "default" cheese in the Anglosphere. The left-hand side of this sequence seems to be based in literal similarity – all three are types of cheese, and two are prominent in Italian cuisine. This may be because all three are surprising in similar contexts (in addition to being literally similar), but it may also be an effect of the non-zero

[3]"Jam" most closely translates to "jelly" or "preserves" in North American English.

[4]"Capsicum" and "icing sugar" are "bell peppers" and "confectioner's sugar" in North American English.

weighting of unsurprising co-occurrences (as controlled by α in Equation 1). The right-hand side is more interesting, and begins to demonstrate the value of s-Glove. Chocolate is similar in context to vanilla, but not as similar as some of the other baking additives. Figs in turn are similar to chocolate, but not as similar as many other confections. What s-Glove provides is that the *combination* of cheddar and chocolate is supposedly like parmesan and vanilla *in terms of why it is surprising*. In other words, s-Glove suggests that someone could be better prepared for the high-surprise combination of parmesan and vanilla following this sequence than by following the literal recommendations of GloVe. We can as of yet offer no validation of this beyond our own opinions. Starting with mozzarella and figs (a common cheese-and-sweet-item combination found often alongside prosciutto, honey, or pistachios) and then proceeding to (cheddar) cheese and chocolate (less common, but still found in baked goods and more adventurous desserts) as a primer for trying parmesan and vanilla seems both plausible and palatable.

Table 8: Surprise walks for *worcestershire sauce and vanilla*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
ketchup	icing sugar	0.25	3.45
paprika	icing sugar	0.25	6.17
worcestershire	*vanilla*	–	*7.82*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
kidney beans	chocolate	0.44	1.19
mustard	ice cream	0.44	4.87
worcestershire	*vanilla*	–	*7.82*

Table 9: Surprise walks for *soy sauce and chocolate*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
ginger	cocoa powder	0.28	3.16
cabbage	cocoa powder	0.28	5.6
soy sauce	*chocolate*	–	*8.55*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
capsicum	jam	0.48	4.87
mushrooms	almond extract	0.43	6.22
soy sauce	*chocolate*	–	*8.55*

Tables 8 and 9 show similar trajectories to the first two examples. Both are combinations of sweet and savoury ingredients, a common theme among highly surprising combinations in our dataset. In Table 8 GloVe again goes with icing sugar as the closest ingredient to vanilla, this time pairing it with ketchup (as in some salad dressings) and then paprika (as in some moderately unusual baked goods). GloVe identifies similarly literal pairings in the soy sauce & chocolate case. S-GloVe, in Table 8 again suggests a sequence of seemingly unconnected but on deeper-inspection flavour-appropriate pairings: mustard ice-cream seems like excellent preparation for whatever unusual recipe could feature worcestershire sauce (a complex and pungent fermented condiment) and vanilla. Beans and chocolate are common combinations in Mexican and Mexican-inspired cuisine, but are still conceptually similar enough to mustard and ice-cream to serve as preparation.

In the final example s-GloVe appears to have selected what is (to the authors) a more unusual and less palatable combination, presented here for the purposes of showing that our preliminary models are far from flawless. Mushrooms are gastronomically quite similar to soy sauce, but the sequence of starting with capsicum and jam, then moving on to mushrooms and almond extract does not, to us, seem as appropriate a preparation for the combination of soy sauce and chocolate. Further developments in the construction of the surprise spaces, the representation of the data, and the algorithm for generating "surprise walks" are needed.

Discussion

In this paper we have presented a proof of concept for how a co-creative system might take planned, sequential action to change human opinion. To our knowledge, this is the first such work, with prior co-creative systems focussing on turn-taking and not conceiving explicitly of longer-term goals. The majority of current interactive creative systems typically do not engage in creative dialogues: they present, re-generate, and present again independently.

The capacity for planned, sequential interactions with creative systems raises a number of possibilities. Systems designed to educate less-expert users could introduce creative artefacts in sequences designed to broaden user horizons. Diverting creators away from low-novelty clusters of artefacts could also be useful outside explicitly educational contexts, given the prevalence of fixation in human designers (Jansson and Smith 1991). Similar approaches have been suggested in data mining contexts as a way to introduce users to the complex nuances of a dataset in an optimal way (Wagstaff et al. 2013). Alternatively, systems designed to diversify the behaviour of their users over time could have benefits for health and nutrition (Grace et al. 2017), using curiosity to overcome orthorexia and neophobia.

Any attempt to influence human behaviour with technology must necessarily be accompanied by an ethical framework. Investigations of what that might entail have arisen from the field of persuasive technology (Berdichevsky and Neuenschwander 1999; Verbeek 2006). Is it right to design systems that seek to change the desires of their users by manipulating their attention and curating their experiences? We, as creativity researchers, can decide that novelty and diversity are worthy of pursuit, but in doing so we implicitly devalue the traditional and the conservative. Luckily, in the contexts we see as near-future applications (education, design and nutrition, for example), it is simple enough to secure user consent in advance. In other contexts, such as using curiosity modelling to customise the news a user consumes, ethical minefields abound.

The most critical next step in this area of research will be to establish how "surprise walks" can be evaluated. The proof-of-concept results in this paper show that the concept has promise, but any further development will require more robust methodologies. One approach would be to devise laboratory experiments in which users are exposed to personalised sequences of artefacts and rate them for novelty, interest, value, etc. This would require "bootstrapping" a user model of knowledge and behaviour in a lab environment. Another approach would be to develop ways to quantify the difference between s-GloVe's "surprise space" and traditional conceptual spaces like GloVe. A final option, and one which remains a long-term goal of our research, would be to develop and evaluate an interactive system for diversifying behaviour by inspiring curiosity through surprise walks.

References

Agres, K.; McGregor, S.; Purver, M.; and Wiggins, G. A. 2015. Conceptualizing creativity: From distributional semantics to conceptual spaces. In *ICCC*, 118–125.

Berdichevsky, D., and Neuenschwander, E. 1999. Toward an ethics of persuasive technology. *Communications of the ACM* 42(5):51–58.

Berlyne, D. E. 1966. Curiosity and exploration. *Science* 153(3731):25–33.

Boden, M. A. 1996. *Dimensions of creativity*. Mit Press.

Clark, E.; Ross, A. S.; Tan, C.; Ji, Y.; and Smith, N. A. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories.

Davis, N.; Hsiao, C.-P.; Singh, K. Y.; and Magerko, B. 2016. Co-creative drawing agent with object recognition. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Davis, N.; Karimi, P.; Maher, M. L.; and Grace, K. 2018. Creative sketching apprentice: Supporting conceptual shifts in sketch ideation. In *Design Computing and Cognition*.

Firth, J. R. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Gärdenfors, P. 2004. *Conceptual spaces: The geometry of thought*. MIT press.

Grace, K., and Maher, M. L. 2014. What to expect when youre expecting: the role of unexpectedness in computationally evaluating creativity. In *Proceedings of the 4th International Conference on Computational Creativity, to appear*.

Grace, K., and Maher, M. L. 2015. Specific curiosity as a cause and consequence of transformational creativity. In *ICCC*, 260–267.

Grace, K., and Maher, M. L. 2016. Surprise-triggered reformulation of design goals. In *AAAI*, 3726–3732.

Grace, K.; Maher, M. L.; Wilson, D.; and Najjar, N. 2017. Personalised specific curiosity for computational design systems. In *Design Computing and Cognition'16*. Springer. 593–610.

Gravina, D.; Liapis, A.; and Yannakakis, G. N. 2016. Constrained surprise search for content generation. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, 1–8. IEEE.

Harris, Z. S. 1954. Distributional structure. *Word* 10(2-3):146–162.

Jansson, D. G., and Smith, S. M. 1991. Design fixation. *Design studies* 12(1):3–11.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Kiddon, C.; Zettlemoyer, L.; and Choi, Y. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 329–339.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. In *FDG*, 213–220.

Linkola, S.; Kantosalo, A.; Männistö, T.; and Toivonen, H. 2017. Aspects of self-awareness: An anatomy of metacreative systems. In *Eighth International Conference on Computational Creativity, ICCC, Atlanta*.

Macedo, L., and Cardoso, A. 2001. Modeling forms of surprise in an artificial agent. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23.

McGregor, S.; Purver, M.; and Wiggins, G. 2016. Words, concepts, and the geometry of analogy. *arXiv preprint arXiv:1608.01403*.

Merrick, K. E., and Maher, M. L. 2009. *Motivated reinforcement learning: curious characters for multiuser games*. Springer Science & Business Media.

Newell, A.; Shaw, J. C.; and Simon, H. A. 1959. *The processes of creative thinking*. Rand Corporation Santa Monica, CA.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Roemmele, M., and Gordon, A. S. 2015. Creative help: a story writing assistant. In *International Conference on Interactive Digital Storytelling*, 81–92. Springer.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. *Computational and Cognitive Models of Creative Design V, Key Centre of Design Computing and Cognition, University of Sydney, Sydney* 113–139.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2(3):230–247.

Swanson, R., and Gordon, A. S. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 2(3):16.

Verbeek, P.-P. 2006. Persuasive technology and moral responsibility toward an ethical framework for persuasive technologies. *Persuasive* 6:1–15.

Wagstaff, K. L.; Lanza, N. L.; Thompson, D. R.; Dietterich, T. G.; and Gilmore, M. S. 2013. Guiding scientific discovery with explanations using demud. In *AAAI*.