Computational Creativity for Valid Rube Goldberg Machines

Xiou Ge
University of Illinois at Urbana-Champaign
xiouge2@illinois.edu

Jinjun Xiong
IBM T.J. Watson Research Center
jinjun@us.ibm.com

Lav R. Varshney
University of Illinois at Urbana-Champaign
varshney@illinois.edu

Abstract

Laboratory activity is an indispensable part of science and engineering education. To develop children's interest in science and engineering, we want to create hands-on activities using artificial intelligence. In this paper, we first describe the use of case-based reasoning (CBR) and an existing knowledge base to yield a combinatorial design space for experiments. We then apply automated planning techniques to generate experiment procedures. We further use functional modeling to represent the experiment devices and demonstrate how that representation enables the planner to generate a valid Rube Goldberg Machine. Finally, a semantic similarity metric is proposed to evaluate the quality of a generated chain of experiments.

Introduction

In Science Olympiad[1] competitions, middle school and high school students from all over the country participate in science experiment design contests to demonstrate relevant scientific concepts. That there are competitions already shows that creating science experiments is not easy. Designing experiments requires not only immense knowledge about the domain but also sufficient information about the properties of available materials. More importantly, students also need imagination and organization skills to arrange the materials rationally and plan out the details of data collection.

Consider building an artificial intelligence system to create novel science experiments. With scientific knowledge and sample experiments in hand, forming useful representations of this data is the key challenge. Much past work has attempted to design experiments for scientific research itself rather than for students. Early work can be traced back to MOLGEN (Stefik, 1981), a knowledge-based system that plans molecular genetics experiments using hierarchical planning techniques. A layered control structure was also introduced to enable meta-planning. MOLGEN focused on the detailed domain knowledge and required much human intervention for a valid experiment plan to be generated. Such systems are not suitable for generating engaging science experiments for students.

Beyond single experiments, it may be more engaging for students to connect a series of devices to form a chain. There is, in fact, a Rube Goldberg Machine (RGM) competition in Science Olympiad called *Mission Possible*[2] for creating chain-reaction machines. The Rube Goldberg Inc also organizes a contest[3] specifically for designing RGM. RGM design has also been brought into class to help teaching. Sharpe, Qin, and Recktenwald (2015) have shown that an RGM-like device setup is good at engaging students and helping them gain deeper understanding of difficult concepts. In fact, Wu et al. (2015) have started to build valid RGMs from the perspective of scene understanding using deep learning and a simulation engine.

In creating such "comically-involved, complicated inventions laboriously contrived to perform a simple operation", judging criteria explicitly require a notion of surprise. As a recent rule book says, "RGMs should work but they also need to capture attention. The more theatrical and funny your machine is, the better it will score!".

In order to build a system that generates creative RGM ideas, we answer several key questions.

- How can knowledge about experimental materials be represented to enable similarity-based retrieval?
- Which class of parts in the existing knowledge base can be used for material substitution?
- How can chains of experiments be generated?
- How can procedure instructions to build RGMs be generated automatically?
- Which generated chain is the most interesting and has highest educational value?

We build algorithmic components to address these questions; putting them together yields a full computational creativity system to generate valid RGMs and assess their quality. By *creativity*, we mean simultaneously achieving novelty and domain-specific quality.

Fig. 1 shows the basic structure of our system. First, we propose a feature-based case representation for experiment materials and adapt mixed-attribute dissimilarity measures from data mining into a distance metric for material retrieval. We also suggest using WordNet to generate more possible substitution materials with the help of word sense

[1] https://www.soinc.org/

[2] https://www.soinc.org/mission-possible-c

[3] https://www.rubegoldberg.com/education/contest/

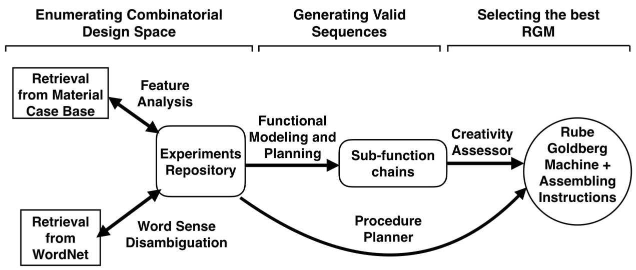


Figure 1: System Structure

disambiguation. Inspired by engineering design, we apply the functional modeling language to represent units used in constructing RGMs and use a forward planner to generate chains of experiments. Procedure plans for building RGMs are also suggested by a partial order planner. We generate examples of experiment chains using our system and propose a creativity evaluation metric for RGMs based on rules from the student competitions and semantic similarity computation using word vectors.

Choosing Materials

Designing science experiments and projects is similar to culinary recipe creation in that both involve suggesting sets of materials and procedures. An AI system with the capability of suggesting unusual combination of materials for a known goal often amuses people (Olteţeanu and Falomir, 2016) and is considered creative according to Bayesian surprise (Itti and Baldi, 2006; Varshney et al., 2013a; França et al., 2016).

In RGM generation, however, not all combinations of devices can be sequenced into a chain due to common material constraints in consecutive devices. Viewing an experimental device as a decomposable system made up of experimental materials, new experiments can be designed if one has access to a set of materials similar to ones in inspiration experiments. By doing this, the constrained combinatorial design space of materials for generating valid RGMs can be enlarged considerably.

Morris et al. (2012); Pinel, Varshney, and Bhattacharjya (2015) suggest that culinary ingredients may be classified into a hierarchy of categories. To generate a recipe, a certain number of ingredients are selected from each category based on pairing rules learned from existing recipes. Unlike in culinary creativity where a single taxonomy of ingredients is applicable to most recipe generation tasks, a suitable classification for one case will likely fail in other cases for experiment generation since the usage of materials is context-dependent.

This issue is more apparent when we try to design experiments using materials that are commonly found at home since a single material may serve different purposes in different scenarios. For example, one might logically classify a marble ball and steel ball into the same category due to their common shape. This would work to roll different objects that perform rotational motion down a ramp and make a series of measurements and observations. However in the Gauss rifle experiment, the marble is not a good substitute for the steel ball since the marble is not ferromagnetic. The marble will not be attracted and accelerate towards the magnet to produce enough momentum to eject the steel bullet.

In addition, a single classification will restrict creativity by dismissing many possible candidates for material substitution. For example, keyboard is put under the computer accessories category whereas wood plank is classified as a type of construction material. In such a taxonomy, keyboard is very distant from wood plank. However, if features such as shape (both approximately cuboid) and surface finish (both have at least one flat surface) are provided, the keyboard will be considered in the set of replacement materials for the wood plank. Therefore, specific feature descriptions of materials are more pragmatic than a comprehensive and refined taxonomy of materials for experiment generation.

Feature-based retrieval

Since science experiment design is knowledge intensive, we want to take advantage of existing data through proper knowledge representation. To ensure the validity of experiment, we start by considering existing experiments as cases and experiment materials as the varying factor.

In engineering design, CBR methods have been applied for material selection (Ashby et al., 2004). Material attributes of mixed types are analyzed and stored in the case base. Based on requirements specified by the designer, a list of materials can be retrieved from the case base. Experiment material substitution is similar to material selection in that features of the original material can be used as key terms to search the knowledge base. Accurate feature information can be extracted from material vendors' websites and refined using crowdsourcing platforms (Demartini et al., 2017). Olteţeanu and Falomir (2016) have demonstrated the effectiveness of feature-based retrieval in creative replacement of everyday objects. Good candidates for replacements are those having high similarity with the original material. In our application, we use the nearest neighbor strategy to search for substitution material.

Material attributes include basic features such as length, shape, or weight, but also context-specific properties such as melting point or electrical conductivity. By referencing the material ontology defined in Ashino (2010), we built a material property ontology to standardize the use of feature names to enable the sharing of material information among different databases using the Protégé ontology editor (Musen, 2015).

Note that material features are not restricted to numeric attributes, but could also include nominal, binary, and ordinal attributes. Han and Kamber (2000) introduced dissimilarity measures for attributes of mixed types. We define the distance metric for our nearest neighbor retrieval in the same manner. Numeric, nominal, binary, and ordinal attributes are dealt with differently as follows. In all equations, x_{if} is the value of attribute f for object i.

- For numeric attributes, the distance is normalized with the difference between the highest and lowest value possible for the particular attribute:

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}} \tag{1}$$

- For nominal or binary attributes:

$$d_{ij}^{(f)} = \begin{cases} 0, & \text{if } x_{if} = x_{jf}. \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

- For ordinal attributes, first count the number of possible ordered states M_f. Then convert the attribute to its corresponding rank, $r_{if} \in \{1, ..., M_f\}$. The rank is normalized and mapped to $[0, 1]$ by the following:

$$Z_{if} = \frac{r_{if} - 1}{M_f - 1} \tag{3}$$

  After conversion, values for ordinal attributes are treated the same way as numerical attributes to compute $d_{ij}^{(f)}$.

- Since not all material features are relevant to a particular experiment, domain experts could label the essential material features to the set E and the less important features to set L. We assign higher weights to more relevant features and lower weights to the less relevant ones when computing the overall distance between material pairs to ensure the replaceability of the retrieved material. The overall distance $d(i, j)$ between experiment material i and j is defined as:

$$d(i,j) = w\frac{\sum_{f\in E}\delta_{ij}^{(f)}d_{ij}^{(f)}}{\sum_{f\in E}\delta_{ij}^{(f)}} + (1-w)\frac{\sum_{f\in L}\delta_{ij}^{(f)}d_{ij}^{(f)}}{\sum_{f\in L}\delta_{ij}^{(f)}} \tag{4}$$

  where $\delta_{ij}^{(f)} \in \{0, 1\}$ indicates whether attribute f appears in both material i and j. $\delta_{ij}^{(f)} = 0$ if an attribute is missing in either material i or j; $\delta_{ij}^{(f)} = 1$ otherwise.

An example of the described knowledge representation and retrieved substitution material is shown in Fig. 2. Material features essential to the problem scenario are highlighted. Constraints are used to check the compatibility of materials within an experiment. The generated combination will be dismissed if materials in a single experiment do not satisfy the constraints specified. Constraints are also used for RGM generation discussed later in this paper where compatibility between different components is essential.

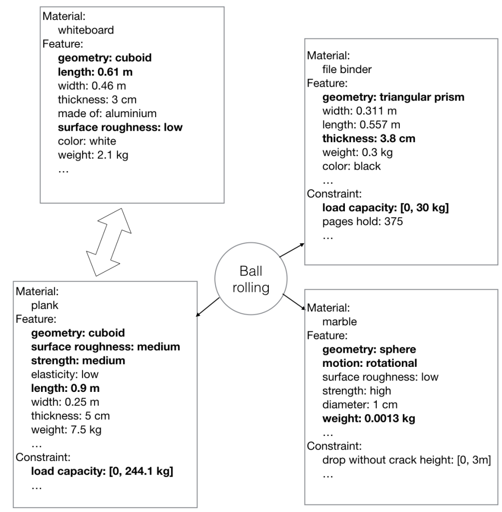


Figure 2: Replacement found by nearest neighbor retrieval

Retrieval from general semantic resources

We propose to augment material substitution retrieval using WordNet (Miller, 1995), a general-purpose knowledge base. In WordNet, nouns are organized into a hierarchical structure in which words are linked by "is a" relationships in terms of their meanings. A more generic concept is referred to as a hypernym whereas a specific instance of a concept is referred to as a hyponym. A hyponym inherits all features of the more generic concept and adds features that distinguishes it from superordinate and sister terms (Touretzky, 1986). Although features of entities are not explicitly specified for each synset entry in WordNet, one can still search for entities with similar features by traversing through the hierarchy. One way of searching is by first looking up the hypernym of the target word and then listing out all hyponyms of the hypernym.

Fig. 3 shows the hierarchical structure in WordNet and selected terms returned on a possible query. In a scenario of building a ramp, terms returned like sheet metal, panel, and plate glass are all good replacements for a board. Also, a query of the hyponym of the target word can also return good candidates like surfboard, ironing board, and wallboard. This method augments the substitution set without requiring extensive human effort in labeling features for experiment materials.

A problem one might face is the material term might be a polysemous word. In WordNet, words are grouped into sets of synonyms called synsets. An example of synset encoding is 'board.n.01', in which the first entry is the word itself, second entry is the part of speech (POS) tag and the third entry represents the index of sense that the term corresponds to. When looking up a word, all possible synsets associated with different meanings of the word will be returned. To search for substitution materials for experiments, the exact synset entry that the original material corresponds to is required. However, the synset entry will not be available unless someone assigns the label manually or using Word Sense Disambiguation (WSD) techniques.

For our application, we use a Support Vector Machine (SVM) classifier to disambiguate the sense of a target word. The training data for the classifier is a list of example sentences that include the target words tagged with corresponding sense labels. We use word embeddings to represent the contextual features since they are more efficient for training and better at capturing relationships among concepts. After the classifier has been trained, it can predict sense labels for previously unseen examples based on the likelihood of each sense given the contextual features (Zhong and Ng, 2010).

As an example, we trained a linear SVM to disambiguate

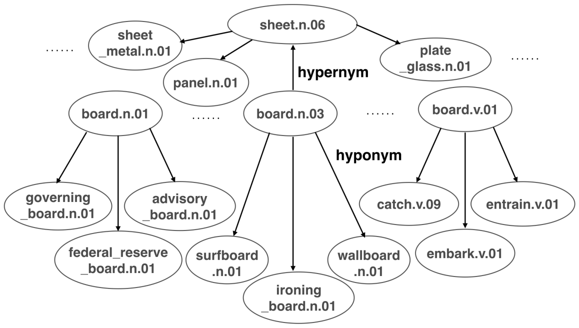


Figure 3: WordNet Hierarchy

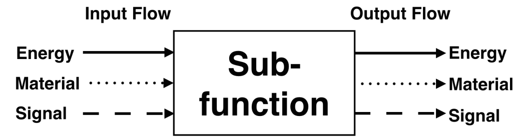


Figure 4: Functional modeling representation of devices

sub-function:	ConvertGPEtoKE
precondition:	Energy(Human)
effect:	Material(Wheel), Energy(K.E.), ¬ Energy(Human)

Table 1: Sub-function Schema

the three most common senses of the word "board". We collected 176 examples in total from several resources[4][5][6] to form a balanced dataset for training. We cleaned up the context corpora by removing punctuations, non-alphabetic characters, and common stop words. Remaining words are converted to their lemma forms in lower case.

A window size of five words on each sides of the target word is used to represent the context. Words within the window are mapped to a list of embeddings $W = \{\mathbf{v}_{-5}, ..., \mathbf{v}_{-1}, \mathbf{v}_1, ..., \mathbf{v}_5\}$. The word embeddings we use are obtained by training the skip-gram word2vec model (Mikolov et al., 2013) available in gensim package (Řehůřek and Sojka, 2010) with Wikicorpus scraped from the science domain. Similar to Iacobacci, Pilehvar, and Navigli (2016), we use the average strategy by computing the centroid of embeddings of the selected surrounding words to obtain the context vector.

$$C = \frac{\sum_{i \in W} \mathbf{v}_i}{|W|} \tag{5}$$

After extracting features for all examples, the set of contextual features and sense label pairs $\{(C_1, S_1), (C_2, S_2), ..., (C_n, S_n)\}$ are used to train the linear SVM. To test the performance of the SVM classifier, we run 5-fold cross-validation on the entire dataset and the accuracy is $M = 0.77, SD = 0.07$.

Rube Goldberg Machine Generation

Experiential learning activities are not limited to the conventional controlled experiment setting where repeated measurements are done to verify certain physical laws or relationships. Instead, learning concepts by building an RGM may be more engaging for students. In an RGM, a series of devices are setup in a way such that one device triggers another in a sequence. Along the chain reactions, many different science and engineering concepts are demonstrated. Learning could be more entertaining if the advisor could suggest possible ideas of building an RGM.

In science projects for students, an experimental setup typically has some function. For instance, a ramp can be considered as a module that performs kinetic energy and potential energy conversion for the object rolling on it. These modules are also frequently used in building an RGM. Given the many possibilities of modules made possible due to combination of materials, the design space will be even bigger if we can build a chain of these modules.

Functional Modeling Representation

Before thinking about automatic chain synthesis, it is important to come up with a systematic way to describe and represent the devices and relationship between them. Given the diversity of devices and the multi-disciplinary knowledge involved, coming up with a consistent knowledge representation is non-trivial. In engineering design, a holistic design is usually disassembled into sub-modules for conceptual analysis. Pahl, Beitz, and Wallace (1984) represent functional modules using block diagrams and call them sub-functions. Each sub-function block has input and output flows that fall in three main categories: energy, material, and signals. Each sub-function can be mapped to a corresponding physical embodiment. As suggested by Bohm, Stone, and Szykman (2005), the functional model allows multiple different types of input and output flows for each block to ensure completeness in knowledge representation. Real mapping examples such as power screwdriver and automobile seat can be found in Hirtz et al. (2002).

Devices in RGMs can be represented using sub-functions. We find the taxonomy of modeling vocabulary defined by Hirtz et al. (2002) useful for representing the devices in an RGM. By referencing the modeling language, we formally analyze the function of each device and its input and output to obtain sub-function representations of device units as shown in Fig. 4. For example, a Gauss rifle device can be interpreted as a system that converts magnetic potential energy to kinetic energy (K.E.). Human effort in the *Energy* category is the input to trigger the system. Both a steel ball in the *Material* category and K.E. in the *Energy* are outputs of the system.

For our application, we represent each sub-function block as planning operators using a STRIPS-like representation. Input and output flows of each sub-function are represented as preconditions and effects associated with the operator re-

[4] http://www.comp.nus.edu.sg/ nlp/corpora.html
[5] http://sentence.yourdictionary.com/Board
[6] http://www.manythings.org/sentences/words/board/1.html

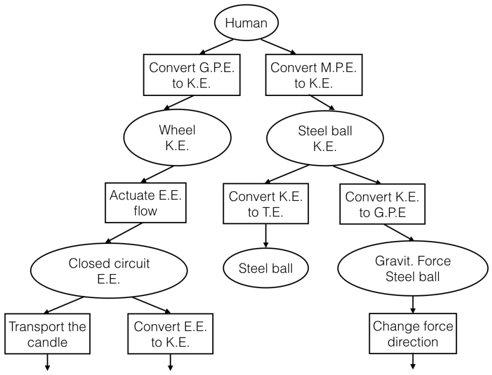


Figure 5: Part of a state space graph

spectively. An example of such a sub-function operator is shown in Table 1. A planning problem can be formed when a set of sub-function operators, initial input, and goal output are specified. We use a forward search algorithm for planning. Part of a possible state space graph expanded by the forward planner is shown in Fig. 5. In the state space graph, each oval represents a world state described by the flows. At first, only the initial state appears in the state space graph. An applicable operator (represented by the rectangles) can be added after a state if its preconditions are supported in that state. A new state is also added to the graph due to the change brought about by the operator. The algorithm terminates when the goal condition is found in a new state. A valid chain of functional blocks is a path in the graph from initial condition to the goal. By mapping each functional block in the chain to its corresponding physical embodiment, we get an RGM. Since materials used in each device are also binded with constraints, these constraints can be used to check the compatibility between adjacent units for physical embodiment selection. Fig. 6 shows several generated sub-function chains and corresponding RGMs.

The idea of building RGMs from science project components based on their input/output matching and compatibility can be further extended to the design of actual engineering systems and products (Li et al., 2013). Knowledge representation in our system is distinct from other planning applications like the story generation (Riedl and Young, 2006). Functional modeling language better reveals the scientific concepts behind the engineering processes and is thus better for educating students.

Suggesting Assembling Procedure Plans

Procedural instructions for building each module and connecting different modules into a chain are equally important. Much research has been done to create procedural artifacts including business processes (Heinrich, Klier, and Zimmermann, 2015), manufacturing simulations, and space missions. In computational creativity, efforts have also been made to create procedural artifacts. In Chef Watson, a graph matching and merging approach has been proposed to create recipe steps (Pinel, Varshney, and Bhattacharjya, 2015). Existing recipe instructions are parsed into directed acyclic graphs in which nodes are ingredients and discrete actions. We find the planning approach appropriate for our system since not all actions in an experiment are associated with a material or a concrete entity.

action:	placeOn(A, B)
precondition:	have(A), have(B), canHandle(A), withinLoadCapacity(A, B)
effect:	on(A, B)

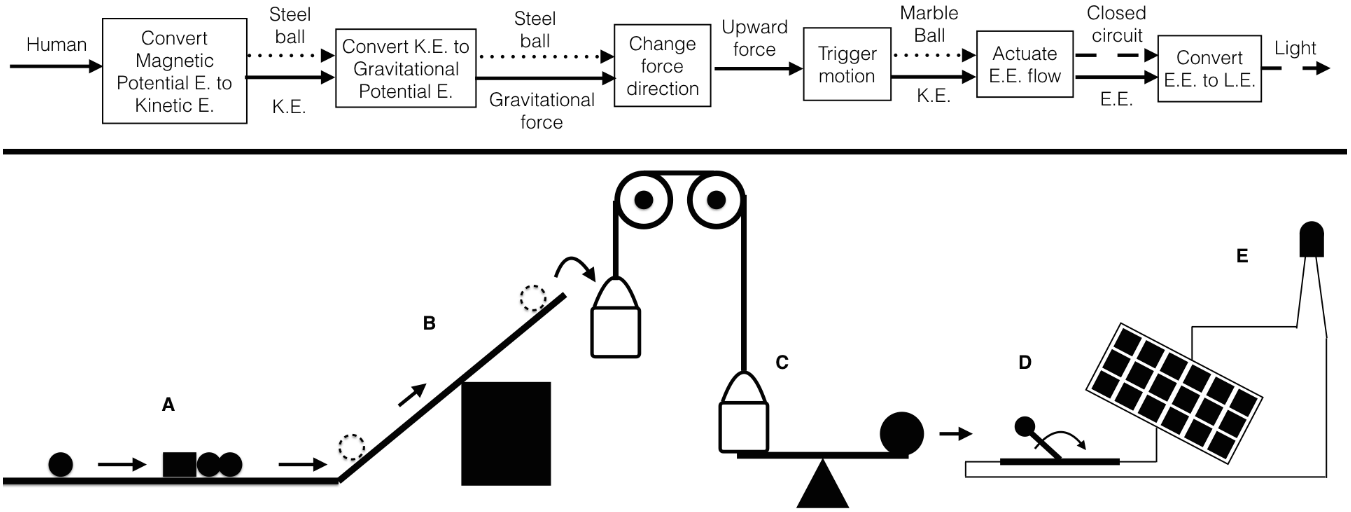
Table 2: Action Schema

For generating procedures, the initial world state can be described with as a set of literals such as available materials, constraints. The desired outcome can be stated as propositions to be satisfied for the goal state. Actions are represented using operators that include a set of preconditions and effects of executing the actions. For our problem, we use a partial order planner to generate plans of assembling procedure. At every iteration, the planner randomly selects an operator from the knowledge base that satisfy any goal conditions, referred to as open condition flaws. Once an action has been instantiated, the preconditions of this action becomes the new open condition flaws. On the next iteration, operators are selected to repair both old and new flaws. A causal link is constructed between action s_1 and s_2 via a specific condition e, represented as $s_1 \xrightarrow{e} s_2$, when execution of s_2 requires condition e established by s_1. The algorithm terminates when each precondition of each action is supported by the effects of a previous action or by conditions in the initial world state. A causal chain of actions that transform the initial world state to the goal state is thus a logical procedure for conducting the experiment. Since partial order planning enforces causal dependency, generated plans are ensured to be valid.

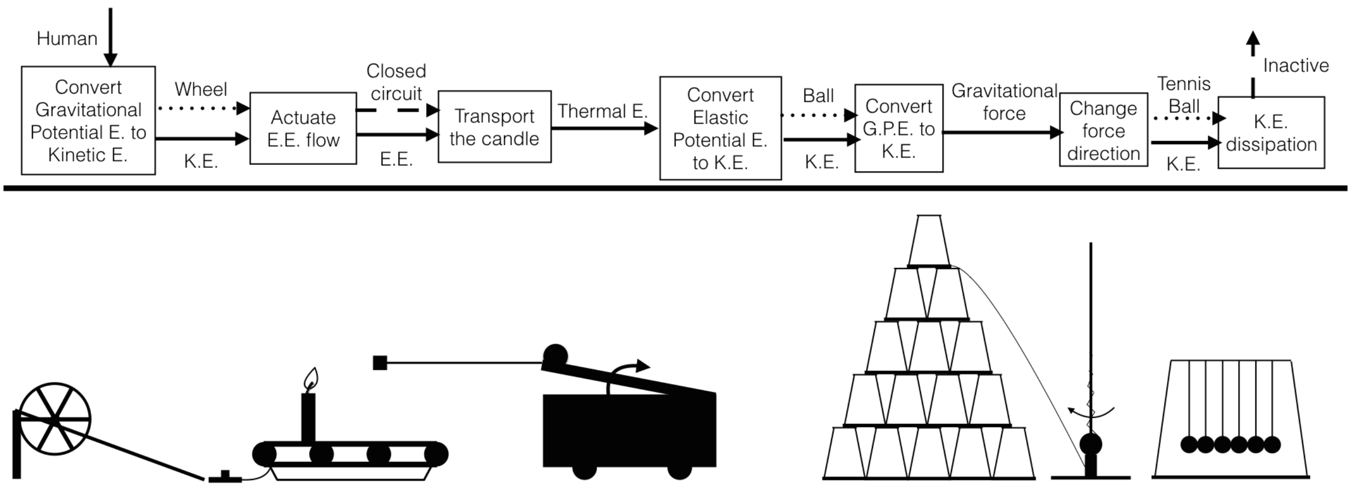
For our problem setting, the STRIPS-like representation is again used to define the planning problem. An example action schema is shown in Table 2. A plan is generated by the partial order planner for the Gauss rifle experiment see Fig. 7. Natural language generation techniques mentioned by Wasko and Dale (1999) can be applied to generate human readable texts from plans.

Selection through Creativity Evaluation

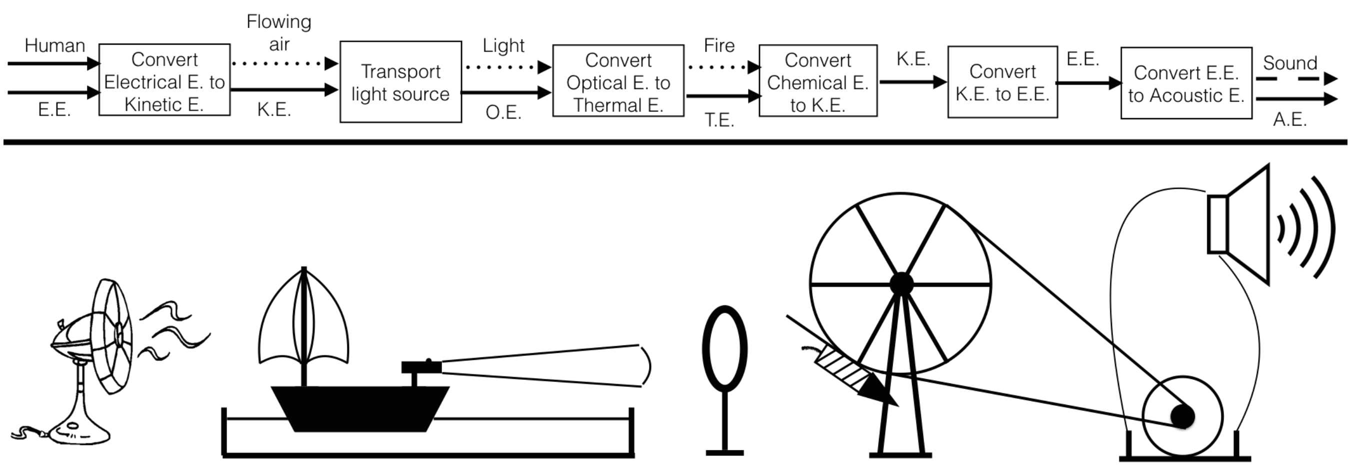
As part of the computational creativity system, internal assessment of creativity of the generated artifact is essential (Varshney et al., 2013b). We find that ramps, spirals, domino, and other physical contact-based devices are very common in RGMs. Creating a machine simply by repeating these components may not be as engaging as those that involve greater variety of reactions. Rules from the “Mission Possible” competition give higher scores to RGMs using components from different categories and having more energy transfers. Considering these rules, we count the number of energy transformations, disciplines, and concepts involved in the three generated RGM examples and display the result in Fig. 8. The example in Fig. 6c might outperform the other two since it involves more energy transfor-



(a) Example a



(b) Example b



(c) Example c

Figure 6: Generated chain of sub-functions and corresponding RGMs

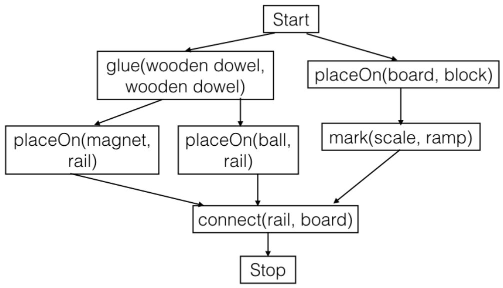


Figure 7: Assembling Plan

Transition	Similarity score
A→B	0.385
B→C	0.337
C→D	0.136
D→E	0.320

Table 3: Cosine score for concepts in each transition

mations and fields of knowledges with comparable number of concepts. We use the above metrics for creativity evaluation since they directly indicate the knowledge content of generated artifacts. According to Cohen (1999), knowledge is one of the three key requirements for creative behavior.

For educational purposes, we think that an engaging chain should demonstrate concepts from different disciplines. In particular, the more different the concepts involved in adjacent devices, the more novel the chain and thus should be given higher priority. We analyzed the concepts involved in each devices; concepts binded with each device in the generated example (a) is shown in Table 4. To measure the extent of transition in concept domains, we need to map each concept to the pre-trained vector representations. Cosine metric has been used to measure the semantic similarity of words in vector representations. We adopt the cosine similarity and compute the transition score by the following:

$$\text{Score} = \frac{(\sum_{i \in C_1} \mathbf{v}_i)^T (\sum_{i \in C_2} \mathbf{v}_j)}{\| \sum_{i \in C_1} \mathbf{v}_i \| \| \sum_{i \in C_2} \mathbf{v}_j \|} \tag{6}$$

where $\mathbf{v}$ is the distributed representation of a concept, C_1 and C_2 are the sets of concepts related to two adjacent components respectively. The transition scores shown in Table 3 agree with our intuition as transitions across different disciplines have lower cosine score than those within a discipline. A chain with low cosine score and more cross-discipline transitions should be considered more creative.

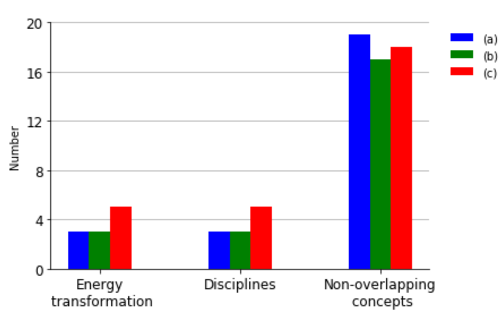


Figure 8: Quality measures

Conclusion

We have described a full computational creativity system that generates RGMs. Several contributions of the system have been discussed. To recap, both CBR and lexical substitution techniques are demonstrated to suggest high quality replacement material. We also apply functional modeling concepts to device representation and generate chains of experiments using a forward planner. Classical planning concepts are applied to represent RGM construction problems and a partial order planner is used to generate procedural instructions. To guide creative artifact selection, we prioritize chains involving the most discipline transitions by computing semantic similarity of relevant concepts. We will continue to develop the system and expand the knowledge base by encoding more components into their corresponding subfunction representations and identifying the related concepts to those components via crowdsourcing.

As future work, we can measure the creativity of generated chains by analyzing the response of human audiences, e.g. through eye-tracking experiments, to understand devices in the chain that are attractors, sustainers, and relators (Candy and Bilda, 2009; Edmonds, Muller, and Connell, 2006). This could potentially help us evaluate the comicality of generated RGMs, which is difficult to measure using which is difficult to measure otherwise.

Acknowledgments

This work was funded in part by the IBM-Illinois Center for Cognitive Computing Systems Research (C3SR), a research collaboration as part of the IBM AI Horizons Network. We thank H. Gong and T. Sakakini for pointing us to the science domain wiki corpus used for the word2vec model.

References

Ashby, M.; Brchet, Y.; Cebon, D.; and Salvo, L. 2004. Selection strategies for materials and processes. *Mater. Des.* 25(1):51–67.

Ashino, T. 2010. Materials ontology: An infrastructure for exchanging materials information and knowledge. *Data Sci. J.* 9:54–61.

Bohm, M. R.; Stone, R. B.; and Szykman, S. 2005. Enhancing virtual product representations for advanced design repository systems. *J. Comput. Inf. Sci. Eng.* 5(2):360–372.

Candy, L., and Bilda, Z. 2009. Understanding and evaluating creativity. In *Proc. 7th ACM Conf. Creat. Cogn. (C&C '09)*, 497–498.

Cohen, H. 1999. Colouring without seeing: A problem in machine creativity. *Special issue on AISB99: Creativity in the arts and sciences* 102:26–35.

A	B	C	D	E
Momentum	Gravity	Torque	Semiconductor	Resistance
Kinetic energy	Conservation of energy	Balance of moments	Resistance	Current
Newton's 2nd Law	Free Fall	Gravity	Closed loop circuit	Voltage
Acceleration	Friction		Power	LED
Magnetism	Newton's 2nd Law			Luminance
Conservation of energy				

Table 4: Concepts involved in each device for example (a)

Demartini, G.; Difallah, D. E.; Gadiraju, U.; and Catasta, M. 2017. An introduction to hybrid human-machine information systems. *Found. Trends Web Sci.* 7(1):1–87.

Edmonds, E.; Muller, L.; and Connell, M. 2006. On creative engagement. *Vis. Commun.* 5(3):307–322.

França, C.; Goes, L. F. W.; Amorim, Á.; Rocha, R.; and Ribeiro da Silva, A. 2016. Regent-dependent creativity: A domain independent metric for the assessment of creative artifacts. In *Proc. Int. Conf. Comput. Creativity (ICCC 2016)*, 68–75.

Han, J., and Kamber, M. 2000. *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann.

Heinrich, B.; Klier, M.; and Zimmermann, S. 2015. Automated planning of process models: Design of a novel approach to construct exclusive choices. *Decis. Support Syst.* 78:1–14.

Hirtz, J.; Stone, R. B.; McAdams, D. A.; Szykman, S.; and Wood, K. L. 2002. A functional basis for engineering design: Reconciling and evolving previous efforts. *Res. Eng. Des.* 13(2):65–82.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proc. Assoc. Comput. Linguist. Annu. Meet. (ACL 2016)*, 897–907.

Itti, L., and Baldi, P. 2006. Bayesian surprise attracts human attention. In *Adv. Neural Info. Process. Syst. 18 (NIPS 2005).* 547–554.

Li, X.; Zhang, Z.-N.; Liu, Z.-L.; and Xie, Y.-B. 2013. A novel semi-heuristic planning approach for automated conceptual design synthesis. *Proc. Inst. Mech. Eng., C J. Mech. Eng. Sci.* 227(10):2291–2305.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Adv. Neural Info. Process. Syst. 26 (NIPS 2013).* 3111–3119.

Miller, G. A. 1995. Wordnet: A lexical database for English. *Commun. ACM* 38(11):39–41.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over beans of pure joy: Culinary ruminations of an artificial chef. In *Proc. Int. Conf. Comput. Creativity (ICCC 2012)*, 119–125.

Musen, M. A. 2015. The protégé project: A look back and a look forward. *ACM AI Matters* 1(4):4–12.

Olteţeanu, A.-M., and Falomir, Z. 2016. Object replacement and object composition in a creative cognitive system. Towards a computational solver of the alternative uses test. *Cogn. Syst. Res.* 39(C):15–32.

Pahl, G.; Beitz, W.; and Wallace, K. 1984. *Engineering Design*. Design Council.

Pinel, F.; Varshney, L. R.; and Bhattacharjya, D. 2015. A culinary computational creativity system. In *Computational Creativity Research: Towards Creative Machines*. Springer. 327–346.

Řehůřek, R., and Sojka, P. 2010. Software framework for topic modelling with large corpora. In *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC 2010)*, 45–50.

Riedl, M. O., and Young, R. M. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Gener. Comput.* 24(3):303–323.

Sharpe, T.; Qin, G.; and Recktenwald, G. W. 2015. A compact device for inductive instruction in general physics. In *Proc. 2015 ASEE Annu. Conf. Expos.*

Stefik, M. 1981. Planning with constraints (MOLGEN: Part 1). *Artif. Intell.* 16(2):111–139.

Touretzky, D. S. 1986. *The Mathematics of Inheritance Systems*. San Francisco, CA, USA: Morgan Kaufmann.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schörgendorfer, A.; and Chee, Y.-M. 2013a. A big data approach to computational creativity. arXiv:1311.1213v1 [cs.CY].

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Schörgendorfer, A.; and Chee, Y.-M. 2013b. Cognition as a part of computational creativity. In *Proc. 12th IEEE Int. Conf. Cogn. Inform. Cogn. Comput. (ICCI*CC 2013)*, 36–43.

Wasko, M., and Dale, R. 1999. Generating natural language descriptions of project plans. In *Advanced Topics in Artificial Intelligence*, 109–121. Berlin: Springer.

Wu, J.; Yildirim, I.; Lim, J. J.; Freeman, B.; and Tenenbaum, J. 2015. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Adv. Neural Info. Process. Syst. 28 (NIPS 2015).* 127–135.

Zhong, Z., and Ng, H. T. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proc. Assoc. Comput. Linguist. Annu. Meet. Syst. Demonstr. (ACLDemos'10)*, 78–83.