User Centered Graphical Models of Interaction

Minsuk Chang
School of Computing
KAIST
minsuk@kais.ac.kr

Juho Kim
School of Computing
KAIST
juhokim@kais.ac.kr

ABSTRACT

In this position paper, I present a set of data-driven techniques in modeling the learning material, learner workflow and the learning task as graphical representations, with which at scale can create and support learning opportunities in the wild. I propose the graphical models resulting from this bottom-up approach can further serve as proxies for representing learnability bounds of an interface. I also propose an alternative approach which directly aims to "learn" the interaction bounds by modeling the interface as an agent's sequential decision making problem. Then I illustrate how the data-driven modeling techniques and algorithm modeling techniques can create a mutually beneficial bridge for advancing design of interfaces.

KEYWORDS

Interaction Modeling, Interface Modeling, Computational Models, Graphical Models, Learning at Scale, Learning in the Wild

INTRODUCTION

There are namely two cultures of data modeling as suggested by Leo Breiman [3], one of which roots to statistical theories targeting mainly inference problems, whereas, the other roots to machine learning targeting mainly prediction problems. Techniques in the former assumes observed data comes from a latent distribution, and recovering the distribution for understanding the observed

CHI 2019, May 4-9, 2019, Glasgow, UK

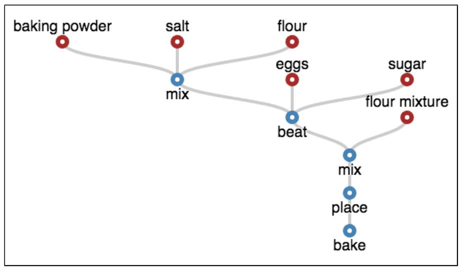

Figure 1: An example of how each recipe can be modeled into a tree structure

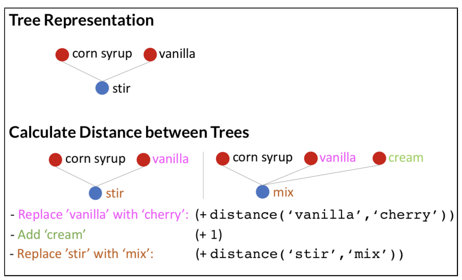

Figure 2: RecipeScape pipeline which uses weighted tree edit distance to calculate structural and semantic similarities between the recipes

phenomena is the key exercise. Techniques in the latter approximates the latent mechanism which results the observed data, and aims to provide a meaningful working model for unseen situations.

This two-culture classification also extends to data-driven interface design. In the data-driven culture, researchers employ statistical methods on interaction data to find patterns for understanding user needs and intents, and build more usable interfaces based on the findings. In the algorithmic culture, researchers use computational models for algorithms to learn and optimize interface performances.

I present examples from the former, data-driven approaches to mining user context and structuring them into a collective semantic representation in various learning scenarios. All three proposed models serve as building blocks for designing interactive systems in creating and supporting learning in the wild. Specifically,

(1) modeling learning (instruction) material to support contextual and nuanced information search.
(2) modeling individual and collective learner workflow to support organic learning opportunities.
(3) modeling user tasks and analyzing them to inform the interaction bounds of an interface.

I also propose an approach from the latter culture, which can directly "learn" the interaction bounds of crowdsourcing tasks by modeling the task interface as a reinforcement learning problem.

MODELING THE LEARNING MATERIAL

The number of available instructional materials even for a single task is easily in the magnitude of thousands, the diversity and the scale of the instructions introduce new user challenges in currently used software interfaces for authoring, sharing and consuming these naturally crowdsourced instructions.

RecipeScape (https://recipescape.kixlab.org) [4, 5] is an analytics dashboard for analyzing and mining hundreds of instructions for a single dish. The computational pipeline (Figure 2) that collects, annotates, computes structural and semantic similarities, and visualizes hundreds of recipes for a single dish uses a graph model for each recipe (Figure 1). Cooking professionals and culinary students found that **RecipeScape** 1) empowers users with stronger analytical capabilities by enabling at-scale exploration of instructions, 2) supports user-centered queries like "what are recipes with more decorations?", and 3) supports creativity by allowing comparative analysis like "where would my recipe stand against the rest?". Moreover, visualization of the computational models allow users to reason and provide their own interpretations and explanations of how the recipes are grouped together in human language, suggesting user with appropriate tools can interpret clustering algorithms. The RecipeScape pipeline illustrates how a graphical representation that captures domain-specific semantics (i.e. cooking) and the structural semantics (procedural instruction) of learning materials enable human interpretable interactions for at-scale analysis and learning.

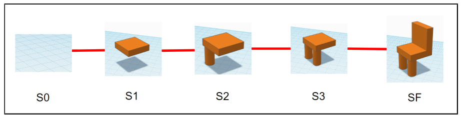

Figure 3: An example user workflow: the state 0 (S0) is the starting blank state, and the user incrementally adds a primitive object, positions it in the correct place and reach state 4 (S4)

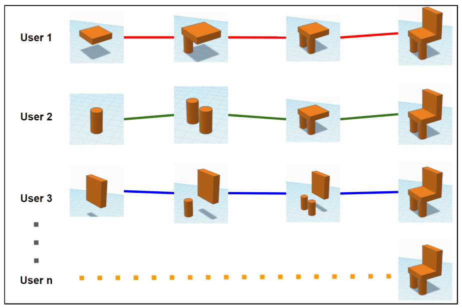

Figure 4: An example illustrating many possible different workflows for the chair modeling task

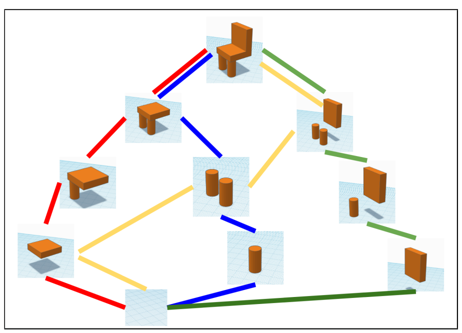

Figure 5: An example of the resulting aggregated representation of the diverse workflows of the chair modeling task

MODELING THE USER WORKFLOW

There are organic learning opportunities in the wild that could impact the way users approach their daily tasks. For example, users of a feature-rich software often want to improve their workflow, for example by learning a shortcut method or learning more suitable methods for a specific post-processing step in 3d modeling.

DemoGraph is a "knowledge graph" constructed from users' demonstrations of 3d modeling tasks captured with screen recordings. It serve as the backbone of an instant user workflow feedback system with peer generated video demonstrations. Each user workflow is represented by a sequence of transitions between "states" of the 3d model as shown in Figure 3. Each state is defined by features that denote the object's progression towards the end goal like the coordinates and the dimension. Each transition is associated with a video snippet segmented from the entire recording of the workflow. The software log which tracks the progression of the 3d model being built automatically segments the videos into the associated video snippets for each state transitions. It captures the sequence of commands, and the CSG(constructive solid geometry) tree of the 3d model. The granularity of the "state" can arbitrarily adjusted, where most fine-grained heuristic is defining each state for every command invocation and the most coarse heuristic is defining each state when every primitive is in the right place, dimensions and color.

The user 3d modeling workflows of the same task all start from a blank state and finishes at the same ending state, making a graph the a natural fit for the resulting representation (Figure 5. The states can be thought of as subgoals in the workflow, making them candidates for potential merge points for user workflows when constructing the graph.

DemoGraph can synthesize a workflow paths that no specific user has exhibited to calculate the fastest path, a workflow that does or does not include a specific command, or the most similar path to a newly submitted workflow.

Similarly, a learnersourcing [10] system can elicit subgoals with problem solving activity with pedagogical values, and the peer generated subgoals can be turned into feedback for other learners in the system [8]. This is particularly useful when the rich software log is not available to determine the states (the progression) of the task.

MODELING THE LEARNING TASK

To understand the interaction bounds an interface poses on users for a task, a learner workflow representation with "semantic" states as subgoals like **DemoGraph** is useful. In a feature-rich software which supports tasks of different complexities and different purposes, some tasks are better supported for learning and some tasks are not. A 3D modeling software supports sketching, modeling, rendering tasks for design, illustration, and 3d printing. For each of the task, we can construct a graph representation of all peer generated workflows as in the previous example of **DemoGraph**. Analyzing the characteristics of the resulting graph constructed from the workflows of each task can inform which tasks are well supported and which tasks are not well supported by the interface, and perhaps which tasks are more "learnable" (i.e. have peer scaffolds by design) and which tasks are not in the interface. For example, compare the task represented by the graph on the *left* in Figure 6 and the task represented by the graph on the *right* in Figure 6. While there are many different possible ways for the task in the right graph, there are only two possible ways for the task in the left graph. If an interface allows diverse workflows for a task, it supports multiple tools and features to reach a shared goal. Such interface provides more scaffolding for the task for learners as more user interaction accumulates.

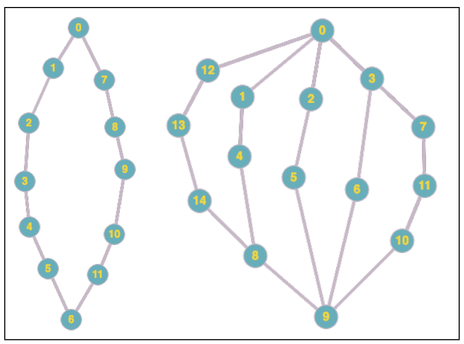

Figure 6: Left: Task with less possible workflows, Right: Task with many possible workflows

MODELING INTERFACES

It is well documented that users behave strategically in the (or lack of) interaction constraints interface gives them. Especially for crowdsourcing tasks, the crowd workers employ strategies to maximize their earnings by investing as little time as possible [7, 9, 11]. The quality of data collected from crowdsourcing tasks are crucial, the vast majority of crowdsourcing task literature has focused on how to do quality control after the data has been collected. However, the there as been very little attempt in trying to understand how an interation design is affecting people to "cheat" under the time-performance tradeoff they're faced with.

Following the line of prior work on learning models of interfaces on a component level [2], on modesl of routine behavior [1], and inferring cognitive models of interaction [6], a crowdsourcing task interface modeling can

A crowdsourcing task interface is the environment with which each crowd worker, an agent, interacts with. Modeling an interface as an MDP(Markov decision process), a model for sequential decision making problems, allows us to computationally calculate what the "optimal behavior" of an agent, the crowdsourcing worker, is given the design of the task interface. Under the bounded rationality perspectives, this interface MDP model and its variants can explain what kind of "bounds" affect crowd workers' behavior, and why they employ certain strategies.

CONCLUSION

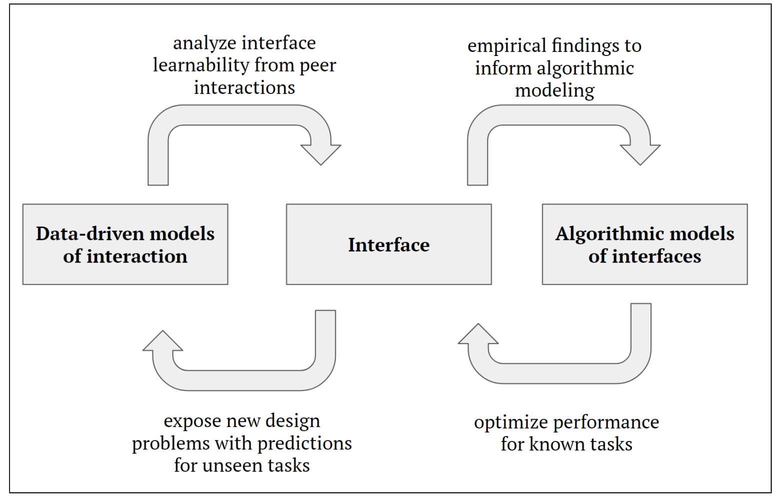

Figure 7: The two cultures of interface modeling and the idealized "interaction model" between the cultures

The goals in HCI systems research are to discover how people use interfaces, heighten our understandings of what makes interfaces usable, and explore novel solutions to usability problems. The bridge is yet to be made between the two cultures, but we can imagine how the two cultures will benefit from each other as illustrated in Figure 7. *"Nowhere is it written on a stone tablet what kind of model should be used to solve problems involving data."* [3]

The outlined and proposed user-centered techniques for structuring graphical models allow data-driven methods to analyze large-scale instructions, suggest workflow improvements, and understand task learnability in an interface. The findings will be valuable ingredients in modeling the mechanisms how interaction with interfaces work. The proposed technique in algorithmic modeling of an interface will lead to diagnosis about tradeoffs people make in an interface, and inform us how to design optimized interfaces for specific tasks. Then we can further optimize of the interface performance for observed interaction scenarios. Also the "optimized" interface can further be evaluated with unseen interaction scenarios, and the data generated will be the ingredients for even further understanding of the interface usability to start a new cycle of design research.

REFERENCES

[1] Nikola Banovic, Tofi Buzali, Fanny Chevalier, Jennifer Mankoff, and Anind K Dey. 2016. Modeling and understanding human routine behavior. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 248–260.

[2] Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2012. Waken: reverse engineering usage information and interface structure from software videos. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 83–92.

[3] Leo Breiman et al. 2001. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science* 16, 3 (2001), 199–231.

[4] Minsuk Chang, Léonore V Guillain, Hyeungshik Jung, Vivian M Hare, Juho Kim, and Maneesh Agrawala. 2018. RecipeScape: An Interactive Tool for Analyzing Cooking Instructions at Scale. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 451.

[5] Minsuk Chang, Vivian M Hare, Juho Kim, and Maneesh Agrawala. 2017. Recipescape: Mining and analyzing diverse processes in cooking recipes. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1524–1531.

[6] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. A cognitive model of how people make decisions through interaction with visual displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM.

[7] Tanya Goyal, Tyler McDonnell, Mucahid Kutlu, Tamer Elsayed, and Matthew Lease. 2018. Your Behavior Signals Your Reliability: Modeling Crowd Behavioral Traces to Ensure Quality Relevance Annotations. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

[8] Hyoungwook Jin, Minsuk Chang, and Juho Kim. 2019. SolveDeep: A System for Supporting Subgoal Learning in Online Math Problem Solving. In *Proceedings of the 2019 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM.

[9] Toni Kaplan, Susumu Saito, Kotaro Hara, and Jeffrey P Bigham. 2018. Striving to earn more: a survey of work strategies and tool use among crowd workers. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

[10] Juho Kim et al. 2015. *Learnersourcing: improving learning with collective learner activity*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[11] Tanushree Mitra, Clayton J Hutto, and Eric Gilbert. 2015. Comparing person-and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1345–1354.