
SolveDeep: A System for Supporting Subgoal Learning in Online Math Problem Solving

Hyoungwook Jin
School of Computing
KAIST
jinhw@kaist.ac.kr

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

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

ABSTRACT

Learner-driven subgoal labeling helps learners form a hierarchical structure of solutions with subgoals, which are conceptual units of procedural problem solving. While learning with such hierarchical structure of a solution in mind is effective in learning problem solving strategies, the development of an interactive feedback system to support subgoal labeling tasks at scale requires significant expert efforts, making learner-driven subgoal labeling difficult to be applied in online learning environments. We propose SolveDeep, a system that provides feedback on learner solutions with peer-generated subgoals. SolveDeep utilizes a learnersourcing workflow to generate the hierarchical representation of possible solutions, and uses a graph-alignment algorithm to generate a solution graph by merging the populated solution structures, which are then used to generate feedback on future learners' solutions. We conducted a user study with 7 participants to evaluate the efficacy of our system. Participants did subgoal learning with two math problems and rated the usefulness of system feedback. The average rating was 4.86 out of 7 (1: Not useful, 7: Useful), and the system could successfully construct a hierarchical structure of solutions with learnersourced subgoal labels.

KEYWORDS

Mathematical Problem Solving;
Learnersourcing; Subgoal Learning; Subgoal Labeling

INTRODUCTION

Mathematical problem solving is not just about mastering individual skills to perform a strategy, but also learning to apply a sequence of appropriate strategies and skills to problems. Learners often struggle in their first attempts at unseen problems because they usually apply the sequence of steps

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5971-9/19/05...\$15.00

<https://doi.org/10.1145/3290607.3312822>

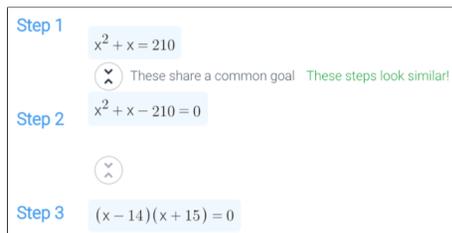


Figure 1: Learners group the steps that share a common goal into a cluster.

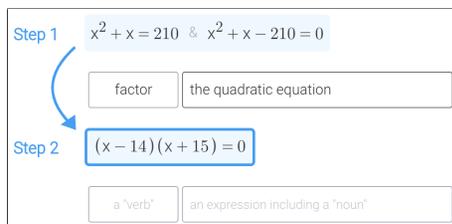


Figure 2: Learners define the goal of each step group while considering the transition from previous steps.

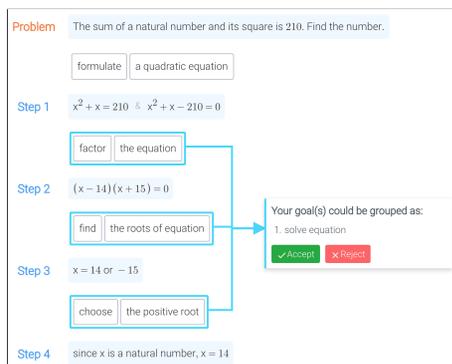


Figure 3: SolveDeep provides feedback on learner-defined subgoals. Learners can accept the feedback to apply the suggestion.

that were successful in previous problems, without appropriate modification of the steps for the new problem context [1]. Subgoal learning has been proven to be helpful because learners form a hierarchical understanding of solution procedure, which is structurally effective in finding the step to modify for a new problem context [1, 3, 4]. Subgoal learning could be supported by labeling steps in a worked example with subgoal labels [3] or by asking learners to generate subgoal labels on their own. The effect of subgoal learning through given labels has been investigated and applied in a wide range of problem solving domains in the context of both classrooms and online learning environments [3, 7, 8].

Although subgoal learning through label generation, compared with seeing given labels, is shown to be more effective to help learners transfer to isomorphic problems [10] and improve their retention [5], it has not been applied at scale in practice because it needs more interactive settings where learners actively generate labels on their own and get feedback on the labels for correction. Since learners are often unable to identify the most pedagogical subgoals for a particular problem domain [1], detailed and interactive feedback on their labels can help learners self-reflect and improve their labels. Developing such at-scale interactive feedback systems has two practical challenges: the shortage of experts who can use authoring tools to develop at-scale feedback systems and a significant amount of time those experts should spend on developing the systems [12].

We use learnersourcing [5] to tackle this problem of developing an interactive system to support subgoal learning in the context of mathematical problem solving. We developed an online learning system, SolveDeep, which presents subgoal labeling activities and guidance with less expert effort to design a feedback system. We reduce the expert effort with a computational pipeline which generates feedback on learner-generated labels by using subgoal labels from previous learners. SolveDeep aims to engage learners in subgoal learning as they label their solutions with subgoals while those subgoal labels are used to power the feedback system for future learners. Many of the previous systems that use a crowd to generate learning materials [5, 6, 14, 15] achieved high quality data by choosing the best data from their collected data set, but SolveDeep has its novelty that it analyzes the complementary structure of its data and aggregate them accordingly to improve the quality of data collection.

To evaluate the feasibility of SolveDeep, we conducted a user study with 7 participants. They were instructed to use the system to group their solution steps, define goals, and revise their goals (Figures 1-3) after solving an algebra problem. All participants found the feedback given by SolveDeep helpful, especially for recognizing missing steps in their own solutions, and reported SolveDeep helped them understand the solution structure better.

Our contributions include:

- SolveDeep, a system that supports learners' subgoal learning in mathematical problem solving with interactive feedback

PROBLEM

The sum of a natural number and its square is 210. Find the number.

GOAL: find x (unknown)

Subgoal: formulate an equation

step 1. $x^2 + x = 210$

step 2. $x^2 + x - 210 = 0$

Subgoal: find the roots of the equation

Subgoal: factorize the equation

step 3. $(x - 14)(x + 15) = 0$

Subgoal: find the values which make the product zero

step 4. $x = 14$ or $x = -15$

Subgoal: choose the positive root

step 5. since the number is a natural number,

step 6. $x = 14$

Figure 4: An example solution to a math problem with a hierarchy of subgoals. Blue labels indicate the mathematical steps used in the solution. A subgoal “find the roots of the equation” can be further decomposed into two subproblems with smaller subgoals. Decomposition of subgoals forms a hierarchical structure. The problem is from a K-12 book for core mathematics [9].

- a computational pipeline that merges complementary sequences of subgoal labels to construct a complete hierarchy of subgoal labels

BACKGROUND

Subgoal Learning. Subgoal learning helps learners form a hierarchical understanding of a solution with subgoals as conceptual components, so that they can deconstruct a problem into subproblems and better recognize the conceptual skeleton of the solution (Figure 4). This hierarchical structure helps learners better understand the solution as they can construct their knowledge from a low level, which involves specific mathematical steps, to a higher level, which describes conceptual procedures. Compared with the flat representation of a solution that learners form when they plainly memorize a step-by-step solution, the hierarchical representation of a solution also provides efficient guidance for modifying steps to solve novel problems [3, 4].

High Quality Subgoal Labels. In this research we define a high quality subgoal label as a goal description that explains the conceptual purpose of a solution step without being bound to problem-specific context (e.g., numbers from a problem). Context-independent subgoal labels are most effective in solving new problems as learners are less tied to the surface features of a problem [1, 3]. We also say a set of subgoal labels is of high quality if they collectively and thoroughly describe a solution procedure and can form a hierarchical representation. We expect our system pipeline to generate these types of high quality subgoal labels and give learners detailed feedback on their labels.

Subgoal Labeling. Subgoal labeling is a technique that labels worked examples with subgoal labels to help learners recognize the procedural structure [2]. We envision learners doing subgoal labeling for the purpose of their own learning. We are investigating a specific form of subgoal labeling, which is learner-driven. While learners passively accept a predefined subgoal structure when they are given subgoal labels, subgoal labeling helps learners to actively self-explain the hierarchical structure of solutions [10]. Since learners generate subgoals, they need feedback on their subgoal labels to correct their misconceptions and to understand the subgoals at varying levels of a hierarchy. This requires feedback to be dependent on the solution and subgoal hierarchy learners initially have.

SYSTEM

SolveDeep aims to support learners in forming a hierarchical understanding on their solution by first defining subgoal labels on their own and then receiving detailed feedback on the hierarchy of their subgoals. Learners solve mathematics problems using the system, and participate in three consecutive activities that follow a typical subgoal labeling activity—grouping solution steps, generating subgoal labels for each group of steps, and revising labels with feedback.

Group Steps (Figure 1). A learner first groups the steps in their solution into clusters that share a common goal. By doing so, the learner forms the basic structure of their conceptual hierarchy. When

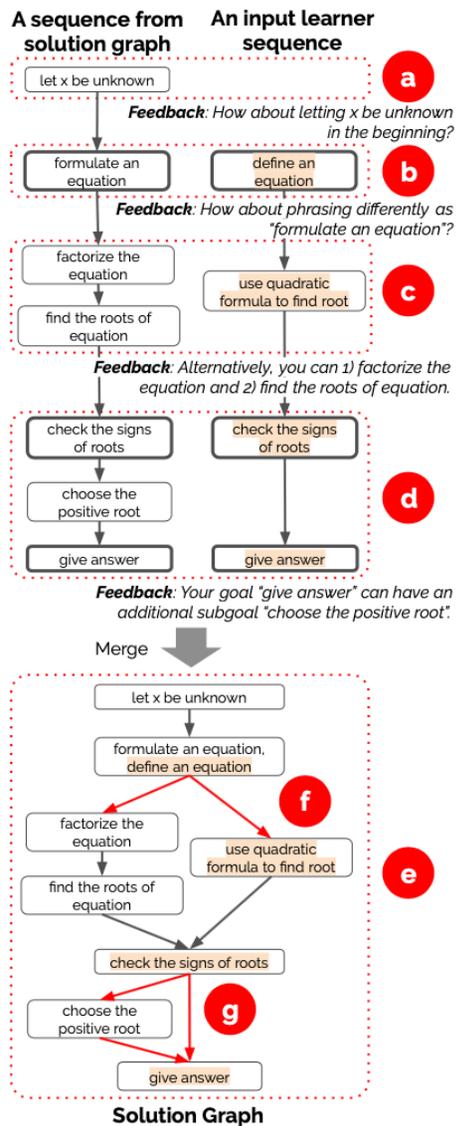


Figure 5: The alignment between two solution sequences can detect: (a) missing subgoals, (b) similar subgoal labels, (c) alternative methods, and (d) higher/lower level subgoals. Nodes with bold border are matching nodes.

the learner tries to group two steps, the system checks text similarity, which is based on the Sørensen-Dice coefficient [13], between the steps and encourages to cluster if the similarity is bigger than 0.6. This heuristic is based on the assumption that the steps with a shared goal have textual similarity because they are consecutive steps. For example, step 1 and 2 in Figure 1 are consecutive steps in formulating an equation. Clustered steps are visually merged into a single step on the interface.

Generate Labels (Figure 2). The learner then defines the goal that each cluster achieves. Learners are encouraged to consider the transition from the previous step when defining the goals. This not only ensures that goals have a logical connection to their previous steps, but also helps learners describe the procedural action taken between steps instead of repeating or plainly rephrasing the mathematical phrases in their solution steps. The system also requires the learner to use a verb and a noun to form a complete clause to ensure they write a meaningful subgoal label. For example, in Figure 2, the learner uses “factor” and “the quadratic equation” to describe the goal of step 2.

Revise Labels (Figure 3). When the learner submits the subgoal labels, the system provides feedback on the content and structure of the submitted labels. The system provides four types of feedback based on different types of alignment patterns ((a) - (d) in Figure 5). The feedback helps learners to either generate more context-independent labels or develop a hierarchical structure by seeing higher/lower level subgoals that can be formed from their subgoal labels. The feedback is shown one at a time, and the learner can accept the feedback to apply it to their subgoal labels. Learners are also allowed to edit the labels directly or go back to previous stages to make changes.

COMPUTATIONAL PIPELINE

To generate a single comprehensive representation of all learner-generated solutions to a specific problem, SolveDeep computes the similarity between two sequences of subgoal labels and their optimal alignment. The alignment is used to detect the hierarchical level differences between the sequences and produce useful feedback that can help learners form a hierarchical understanding. This pipeline reduces the expert effort to build an at-scale feedback system as it can quickly populate different solution strategies with scale.

Solution Graph. A subgoal hierarchy is represented as a graph, in which nodes are clusters of similar subgoal labels and edges represent the sequential order of subgoals in a solution (Figure 5 (e)). A solution procedure is represented as the path along directed edges from the source node. Branching edges (f) represent different solution strategies, and the transitive edges (g) represent the substructure of a goal in a hierarchy. In this graph, we aim to have enough nodes and edges to describe multiple possible ways to solve a problem and represent a hierarchical structure of subgoals.

Computing the Similarity and Optimal Alignment. The similarity between two sequences of subgoal labels is determined by the optimal alignment of the sequences. The optimal alignment is computed by the global sequence alignment algorithm [11] with the scoring scheme of {Match:

Problem: A gardener has 40 m of fencing to enclose a rectangular garden plot, where one side is an existing brick wall. Suppose the two new equal sides are x m long. Find the length of the longer side of the garden that has the maximum area.

Figure 6: The second problem given to the participants. The problem is from a K-12 book for core mathematics [9].

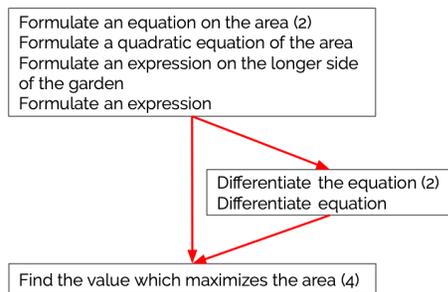


Figure 7: The transitive edge shows a hierarchical structure. “Differentiate equation” is the lower level subgoal of “Find the value which maximizes the area”. The numbers in bracket indicate the number of same labels in a cluster.

the textual similarity score between subgoal labels, *Insertion/deletion*: 0, *Mismatch*: -0.2}. The textual similarity is calculated based on the Sørensen-Dice coefficient [13] and is normalized to a 0-1 range.

Applications. There are two applications of this computation: merging an input sequence to the right position in a solution graph and generating feedback based on alignments. When a learner submits a sequence of subgoal labels, our system chooses, from its solution graph, the sequence with the highest similarity. The system performs a graph union operation to merge the sequence with the graph; the input subgoal label which has a matching node in the optimal alignment is added to the matching node, and the labels with no match are added as new nodes. The connections between the labels in the input sequence are added as edges between corresponding nodes in the solution graph. For example, in Figure 5, two sequences are merged into a solution graph based on their alignments.

The alignment algorithm is also used to detect and generate feedback on the structure of an input sequence. The misalignments in two sequences of labels (Figure 5 (a)-(d)) imply structural differences, which can produce useful feedback on structure. For example, in the case (d) where there are two aligned labels and one additional label on one side, the additional label could be inferred as the lower level subgoal of the second aligned label. Learners then get the feedback that their subgoal could be decomposed into lower level subgoals. To provide rich structural feedback from an alignment, SolveDeep intentionally chooses the sequence with partial similarity to an input sequence.

EVALUATION

We recruited 7 undergraduate and graduate students at KAIST to evaluate the usefulness and effectiveness of SolveDeep in producing a high quality set of subgoal labels. Participants were asked to solve two algebra problems using SolveDeep. The first problem was given to help participants familiarize themselves with the interface and the workflow; we did not analyze the data generated from the first problem. For the second problem (Figure 6), the system initially had a solution graph constructed from a single subgoal sequence generated by an external participant.

The solution graph resulting from the study has 22 nodes in total, of which 9 nodes have a cluster of multiple subgoal labels. The solution graph presents hierarchical structures of subgoals as shown in Figures 7 and 8. Among 39 subgoal labels collected, 30 subgoal labels are context-independent (e.g., “define an equation based on given conditions.”, whereas 9 subgoal labels are operation-specific (e.g., “divide equation by 4”). SolveDeep can provide high quality labels selectively among these labels as it uses the feedback acceptance rate to filter low quality labels in a cluster. The rate could be also used to monitor the quality of a solution graph and require expert interventions when needed.

Participants rated the usefulness of system feedback on their subgoal labels using a 7-point Likert scale (1: Not useful, 7: Useful). The average rating was 4.86 (min: 4, max 6). Participants especially liked the system’s suggestion to add implicit steps they missed in their solution and felt the feedback was detailed and appropriate. Some participants felt the feedback on alternative methods was not

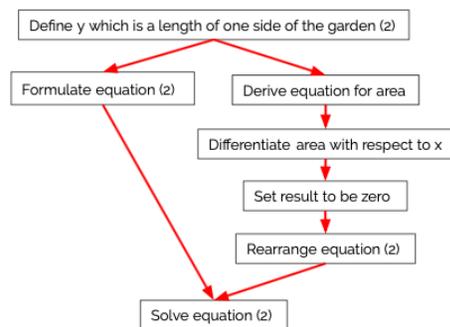


Figure 8: The structure shows how the subgoal “Formulate equation” can be decomposed into multiple lower level subgoals.

helpful for revision as the methods were not directly related to their strategies. We expect a more mature solution graph with more learners’ input could resolve the problem of cold start and provide more fruitful feedback to participants. We leave this as our future work.

ACKNOWLEDGMENTS

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

REFERENCES

- [1] Richard Catrambone. 1995. Aiding subgoal learning: Effects on transfer. *Journal of educational psychology* 87, 1 (1995), 5.
- [2] Richard Catrambone. 1996. Generalizing solution procedures learned from examples. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 22, 4 (1996), 1020.
- [3] Richard Catrambone. 1998. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General* 127, 4 (1998), 355.
- [4] Richard Catrambone and Keith J Holyoak. 1990. Learning subgoals and methods for solving probability problems. *Memory & Cognition* 18, 6 (1990), 593–603.
- [5] Juho Kim. 2015. *Learnersourcing: improving learning with collective learner activity*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- [6] Ching Liu, Juho Kim, and Hao-Chuan Wang. 2018. ConceptScope: Collaborative Concept Mapping for Video Learning. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 387.
- [7] Lauren E Margulieux and Richard Catrambone. 2014. Improving problem solving performance in computer-based learning environments through subgoal labels. In *Proceedings of the first ACM conference on Learning@ scale conference*. ACM, 149–150.
- [8] Lauren E Margulieux, Mark Guzdial, and Richard Catrambone. 2012. Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications. In *Proceedings of the ninth annual international conference on International computing education research*. ACM, 71–78.
- [9] David Martin. 2012. *Mathematics for the International Student: Mathematics HL (Core)*. Haese Mathematics.
- [10] Briana B Morrison, Lauren E Margulieux, and Mark Guzdial. 2015. Subgoals, context, and worked examples in learning computing problem solving. In *Proceedings of the eleventh annual international conference on international computing education research*. ACM, 21–29.
- [11] Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48, 3 (1970), 443–453.
- [12] Andrew M Olney and Whitney L Cade. 2015. Authoring intelligent tutoring systems using human computation: designing for intrinsic motivation. In *International conference on augmented cognition*. Springer, 628–639.
- [13] Thorvald Sørensen. 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.* 5 (1948), 1–34.
- [14] Sarah Weir, Juho Kim, Krzysztof Z Gajos, and Robert C Miller. 2015. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 405–416.
- [15] Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. 2016. Axis: Generating explanations at scale with learnersourcing and machine learning. In *Proceedings of the Third (2016) ACM Conference on Learning@ Scale*. ACM, 379–388.