IIAI Open Conference Publication Series
IIAI Letters on Business and Decision Science
Vol. 005, LBDS474
DOI: https://doi.org/10.52731/lbds.v005.474

An Extension of the Levenshtein Distance for Learning Process Analysis in Constructive Learning and an Evaluation of its Effectiveness

Natsumi Tanabe [*], Karin Yagami [*], Shimpei Matsumoto [*]

Abstract

In this study, we propose an applied extension of the Levenshtein distance for analyzing learning logs in constructive programming learning. By incorporating the influence of blank and dummy cards—which conventional distance metrics fail to adequately capture—our method enables a more detailed evaluation of the quality of learners' trial-and-error processes and the presence or absence of logical thinking. This approach clarifies the distinction between exhaustive trial behaviors and logically constructed problem-solving, thereby contributing to learning support systems that can provide more personalized and appropriate feedback.

Keywords: Constructive Programming Learning, Learning Log Analysis, Levenshtein Distance, Logical Thinking.

1 Introduction

Constructive learning is a learning style in which learners deepen their understanding through trial and error, and it has been attracting particular attention in programming education[(1)]. In recent years, the importance of learning analytics (LA), which utilizes process data (log data) in such learning, has been increasing, and the author has also proposed methods for quantitatively visualizing and analyzing the learning process[(2)].

However, in conventional learning log analysis, the difference between the learner's answer and the correct solution has typically been measured using string edit distances, particularly the Levenshtein distance. Nevertheless, elements such as blanks in the answer field and dummy cards unnecessary for the correct solution have not been sufficiently considered. As a result, it has been difficult to distinguish whether a learner has logically understood the program structure and placed the cards accordingly, or whether they are simply inserting cards at random.

The purpose of this study is to improve the accuracy of learning process analysis by applying an extended version of the Levenshtein distance that reflects the effects of blanks and dummy cards on card operation logs in constructive programming learning support systems. Specifically, this

[*] Hiroshima Institute of Technology, Hiroshima, Japan

study proposes two new distance measures: the "blank-aware distance," which treats unplaced sections as zeroes, and the "dummy card correction distance," which imposes penalties based on the number of dummy cards inserted. This paper clarifies the kinds of learning characteristics that can be visualized through comparison with conventional methods.

The structure of this paper is as follows. Chapter 2 reviews related work and organizes prior research on the Levenshtein distance. Chapter 3 details the definitions and implementations of the proposed blank-aware distance and dummy card correction distance. Chapter 4 reports the experimental setup and evaluation results, and Chapter 5 discusses the findings based on the results. Finally, Chapter 6 summarizes the conclusions and discusses future challenges.

2 Related Work

In the fields of Learning Analytics (LA) and Educational Data Mining (EDM), numerous methods have been proposed to visualize and analyze learning processes by utilizing time-series data obtained from learners' operation logs. Papamitsiou et al. (2014) systematically reviewed empirical studies in LA and EDM and clarified that the discovery of learning paths, clustering, and the construction of predictive models based on log data have been actively pursued[(3)].

Meanwhile, edit distance has attracted attention as a method for comparing and evaluating learners' behavioral sequences. Boroujeni and Dillenbourg (2019) clustered learners' operation logs using the Levenshtein distance, extracting groups of learners who exhibited similar behavioral patterns, and demonstrated the potential for classifying learning strategies and issuing early warnings[(4)].

The Levenshtein distance is defined as the minimum number of edit operations (insertions, deletions, and substitutions) required to transform one string into another. Since its proposal by Vladimir Levenshtein in 1966, it has been widely studied and applied[(5)]. The Wagner–Fischer algorithm (Wagner and Fischer, 1974) based on dynamic programming is commonly used for its computation, with a time complexity of O(mn) for string lengths m and n. Furthermore, various extensions have been proposed, such as the Damerau–Levenshtein distance, which allows the transposition of adjacent characters (Damerau, 1964), and weighted edit distances, which assign different costs to different operations, thereby expanding applications from text processing to bioinformatics[(6)(7)].

However, in card operation logs within programming learning support systems, domain-specific elements such as "unselected blanks" and "dummy cards" emerge. Conventional edit distances often fail to accurately capture these elements when treating them as simple strings, leading to a loss of critical information and making it difficult to precisely evaluate the quality of learners' trial-and-error processes and their level of understanding. In this study, we propose two new distance measures—"blank-aware distance" and "dummy card correction distance"—to explicitly reflect these elements and aim to improve the evaluation accuracy in learning process analysis.

3 Methodology

3.1 COPS System Overview and Log Data

The Card Operation-Based Programming Learning Support System (hereinafter referred to as COPS) targeted in this study is an exercise support system in which learners are presented with multiple program cards and are required to rearrange them to construct correct source code. Each card contains one or more lines of source code, and in addition to the cards necessary for the correct solution, dummy cards intended to induce incorrect answers are also included. Furthermore, the number of cards that can be placed in the answer field is fixed in advance. Therefore, learners must appropriately arrange the necessary cards in the correct order while avoiding the dummy cards (see Figure 1).

Learners can receive feedback on their answers—such as the correctness of each card placement and the presence of dummy cards—as well as hints related to solving the problem by pressing the "submit answer" button. Answers can be submitted multiple times for a single problem, and learners are allowed to abandon a problem midway and proceed to the next problem if they wish.

Additionally, all learner operations are recorded as logs. These logs include not only the history of card operations but also the records of answer submissions and the contents of the feedback. Each card is assigned a unique number, and in the logs, the history of card operations is stored as a sequence of integers.

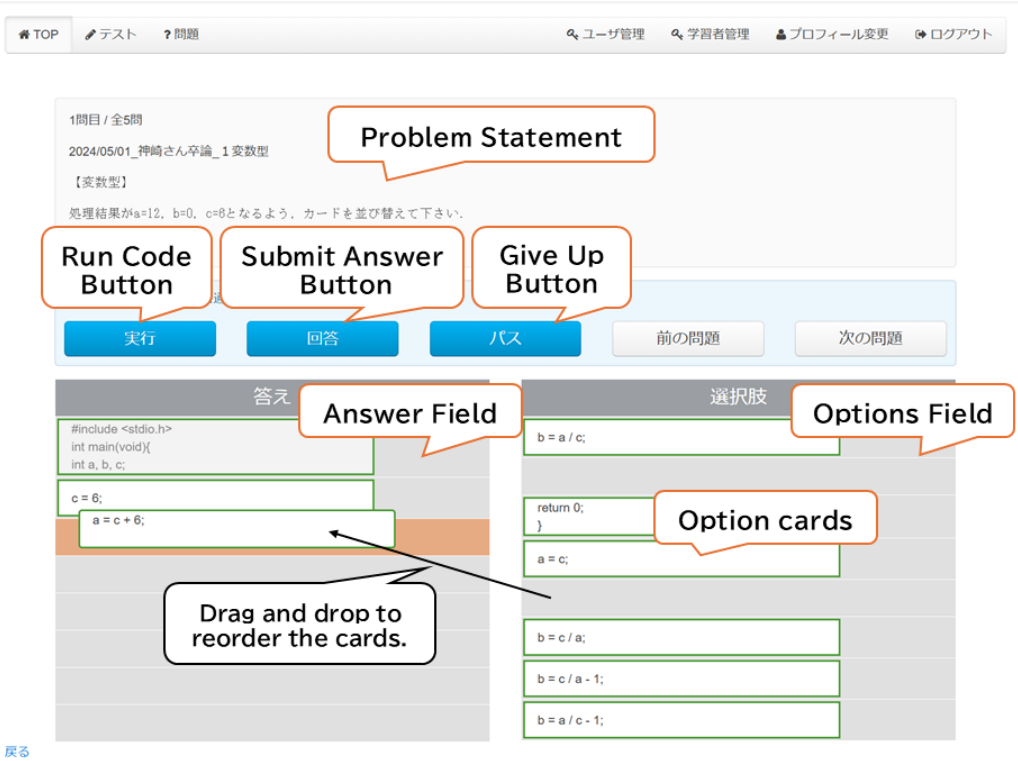

Figure 1: Exercise Screen of COPS

3.2 Standard Levenshtein Distance

Levenshtein Distance is a distance metric defined as the minimum total number of edit operations (insertion, deletion, substitution) required to transform one sequence into another. Formally, it satisfies the following metric conditions:

(1) **Non-negativity**: $d(x,y) \geq 0$

(2) **Identity**: $d(x,y)=0 \Leftrightarrow x=y$

(3) **Symmetry**: $d(x,y)=d(y,x)$

(4) **Triangle Inequality**: $d(x,z) \leq d(x,y)+d(y,z)$

The calculation is performed using dynamic programming based on the Wagner–Fischer algorithm, which can compute the distance between sequences of length O(mn) time. In the COPS log analysis, the sequence of cards in the learner's answer field and the correct sequence of cards are treated as strings, and by calculating the Levenshtein distance, the proximity of the learner's answer to the correct one at that point in time is quantitatively evaluated.

3.3 Extended Distance Metrics

In the traditional Levenshtein Distance, elements specific to COPS, such as empty fields in the answer area or dummy cards, are sometimes ignored. Therefore, in this study, four extended metrics are defined as follows:

(1) levenshtein_distance: This applies the standard Levenshtein distance, where empty fields in the answer area are removed from the string before calculation.

(2) modified_levenshtein: In addition to the standard distance, the distance value is increased by 1 for each occurrence of a dummy card in the answer area.

(3) levenshtein2: Empty fields in the answer area are treated as a special symbol "0", and the Levenshtein distance is calculated between sequences that include this "0".

(4) modified_levenshtein2: In addition to the "levenshtein2" metric, 1 is added to the distance value for each occurrence of a dummy card.

These extensions do not change the original metric definition but rather reflect information about empty fields and dummy cards in the COPS logs as practical metrics through preprocessing and correction. In the following, the metrics defined in this section are used to visualize and evaluate the learner's trial-and-error process.

4 Experiment

This chapter describes the experiments conducted to demonstrate the effectiveness of the proposed method. Specifically, learning logs from a programming exercise conducted in a university course are used, and the proposed extended Levenshtein distance is applied to analyze the

learning process. The results of this analysis are presented.

4.1 Dataset and Experimental Settings

In this study, programming course data from first-year students in the Intelligent Information Systems major, conducted in the 2016 academic year, were used. The target group consisted of 105 students, and exercises using the COPS system were conducted in 7 out of the 15 lectures. In each exercise session, after a lecture on basic content, card operation-based tasks were presented to assess understanding. The tasks were in the form of a test, with each session including one major question and two to three minor questions. Talking during the exercises was prohibited, and each task had a 10-minute time limit [8][9].

From the exercise tasks, "Problem 34" (a task related to nested loops) was selected, and the analysis results of the operation log data from three learners (user16, user36, user74) are presented. Figure 2 shows the content of Problem 34. In this problem, six correct cards are provided, and in the log data, these are represented by positive integer numbers (1-6). These numbers correspond to the correct order of the cards.

Additionally, to induce incorrect answers from learners, dummy cards (card number -1) are also included. Figure 3 shows the card operation history for each learner.

Problem 34_Nested loop	No.	Option cards

Rearrange the cards so that the output consists of '*' arranged in 4 rows and 2 columns.
 	
#include <stdio.h>
int main(void){
int i, j ;
//Rearrange the code inside this section
return 0;
}
	1	for(i=0; i<4; i++){

	2	for(j=0; j<2; j++){
	3	printf("*");
	4	}
	5	printf("¥n");
	6	}
	-1	printf("*¥n");
	-1	for(i=0; i<=4; i++){
	-1	for(j=0; j<=2; j++){

Figure 2: Problem Statement and Choice Cards for Problem 34

turn	user16	user36	user74
1	[2][][][][][]	[][1][][][][]	[-1][][][][][]
2	[2][3][][][][]	[][1][-1][][][]	[-1][3][][][][]
3	[2][3][6][][][]	[][1][-1][3][][]	[-1][3][4][][][]
4	[2][3][6][1][][]	[][1][-1][3][4][]	[-1][3][4][-1][][]
5	[2][3][6][1][-1][]	[][1][-1][3][4][-1]	[-1][3][4][-1][-1][]
6	[2][3][6][1][-1][4]	[1][][-1][3][4][-1]	[-1][3][4][-1][-1][6]
7	[2][][6][1][-1][4]	[1][-1][][3][4][-1]	[-1][3][][-1][-1][6]
8	[2][1][6][][-1][4]	[1][-1][3][][4][-1]	[-1][3][4][-1][-1][6]
9	[2][1][][][-1][4]	[1][-1][3][4][][-1]	[][3][4][-1][-1][6]
10	[2][1][3][][-1][4]	[1][-1][3][4][-1][]	[][3][4][][-1][6]
11	[2][][3][1][-1][4]	[1][-1][3][4][-1][6]	[-1][3][4][][-1][6]
12	[2][3][][1][-1][4]	[1][-1][3][4][][6]	[-1][3][4][-1][-1][6]
13	[2][3][1][][-1][4]	[1][-1][3][4][5][6]	
14	[2][3][1][-1][][4]	[1][][3][4][5][6]	
15	[2][3][1][-1][6][4]	[1][2][3][4][5][6]	

Figure 3: Logs of Three Learners Solving Problem 34

4.2 Evaluation by Distance Metrics

In this section, we present graphs that visualize the changes in distance over time by applying both the standard Levenshtein distance and the three proposed extended distances to the operation logs of three learners (see Figures 4 to 6).

In the graphs, yellow represents the standard distance, blue represents the dummy card-corrected distance, green represents the blank-considered distance, and purple represents the blank plus dummy-corrected distance. Red markers indicate the timing of answer submissions.

The starting point of the time axis in each graph is standardized to the moment when the first card is inserted (0 seconds).

Additionally, to complement the explanation, a part of the card operation history shown in Figure 3 is reconstructed to visually clarify the transitions in the actual answer fields, as shown in Figures 7 to 9.

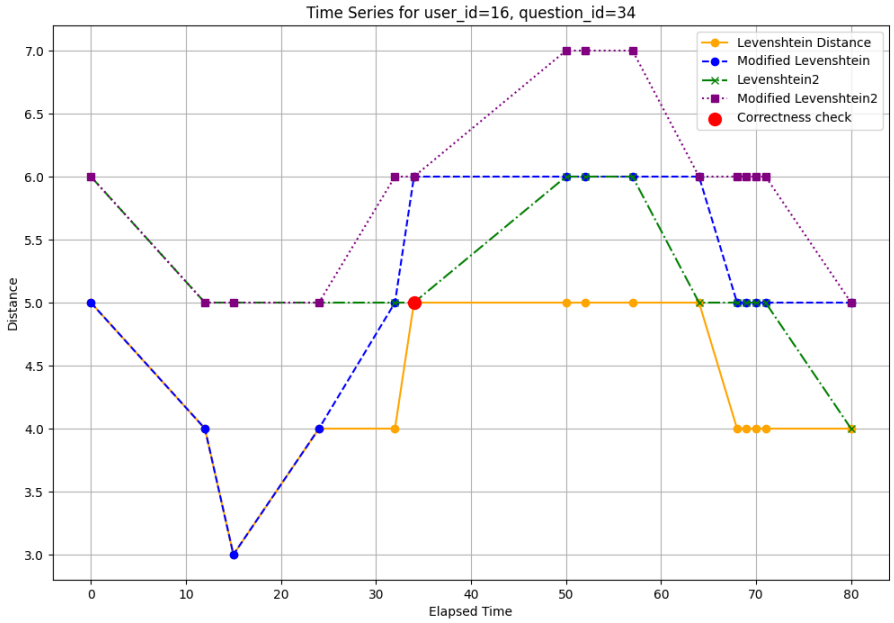

Figure 4: Time-Series Transition of Distance (user16)

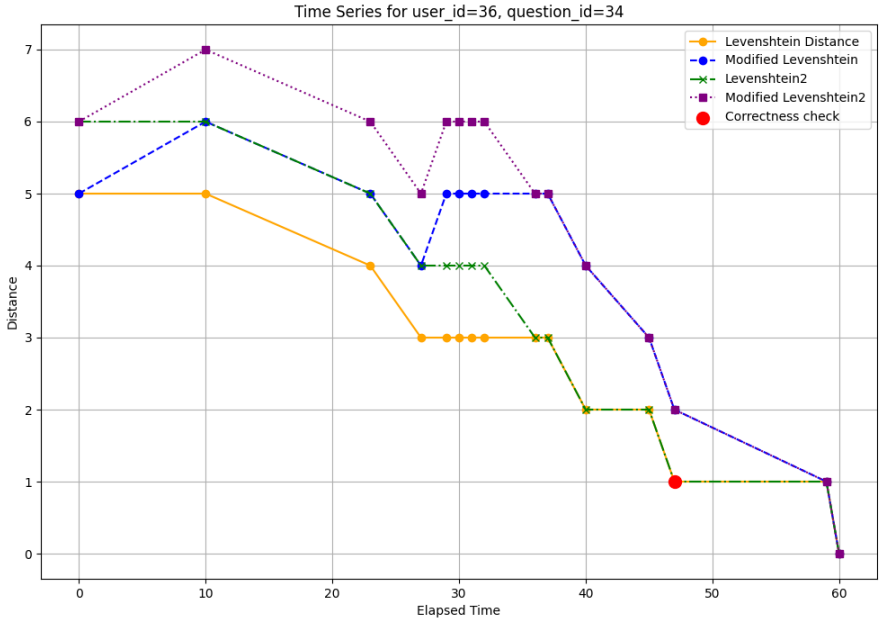

Figure 5: Time-Series Transition of Distance (user36)

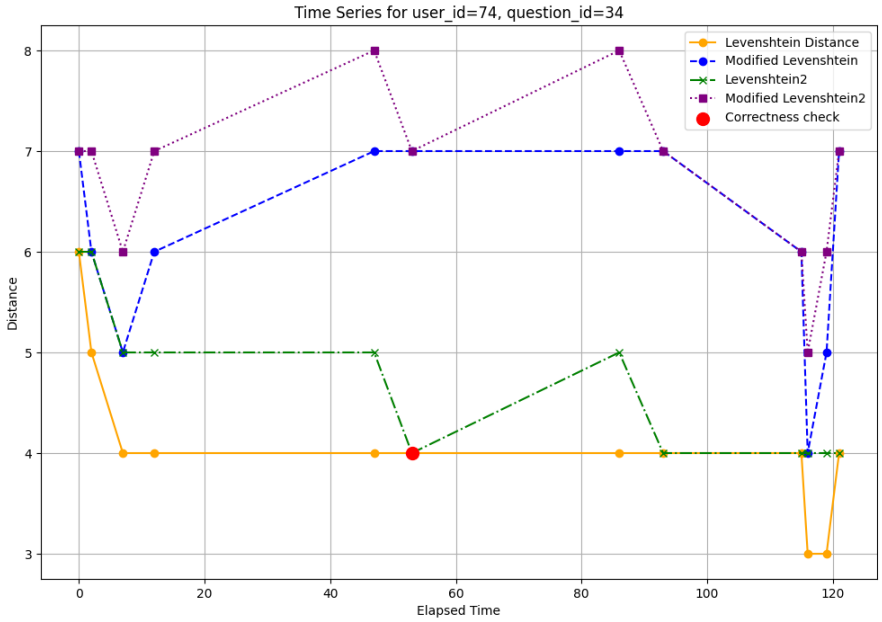

Figure 6: Time-Series Transition of Distance (user74)

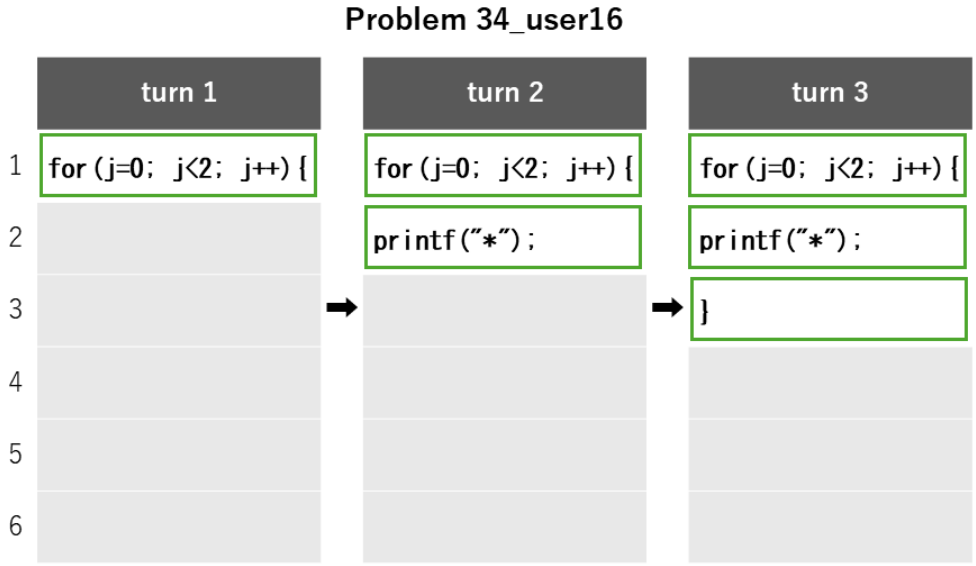

Figure 7: User 16: Answer field transition (partial view)

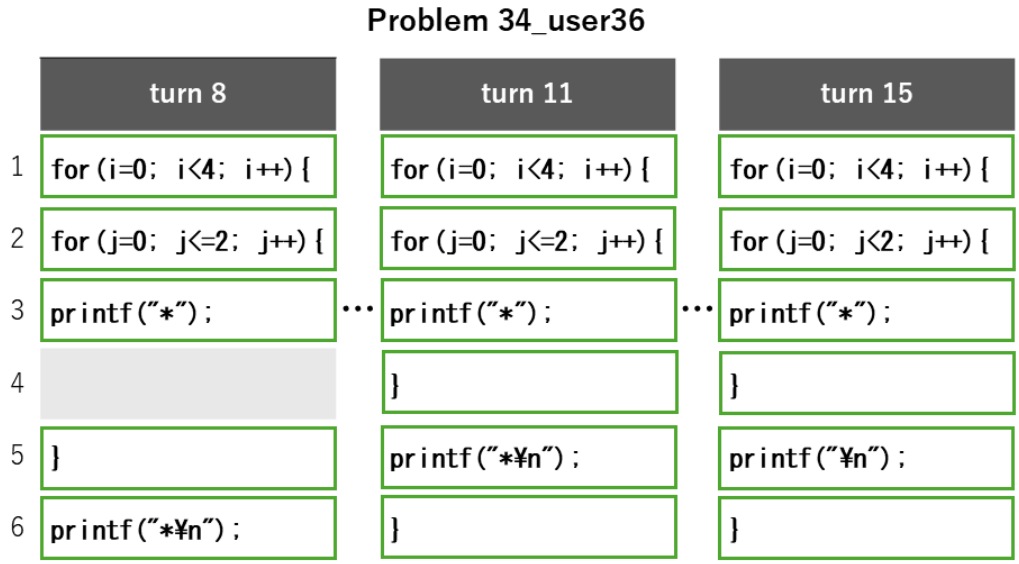

Figure 8: User 36: Answer field transition (partial view)

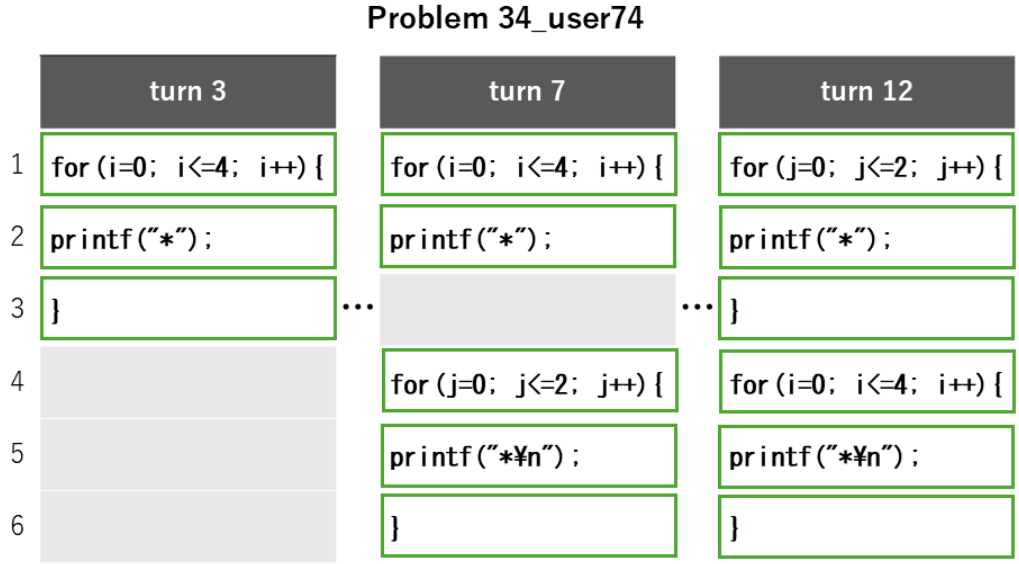

Figure 9: User 74: Answer field transition (partial view)

First, we focus on the learning process from turn 1 to turn 3 of participant user16.
Based on the standard Levenshtein distance, the distance decreases from 6 to 3, suggesting that the learner is steadily approaching the correct answer.
However, when checked using the blank-considered extended distance, the distance only slightly decreases from 6 to 5, indicating a slower learning progression.
As shown in Figure 7, at turn 1, user16 inserted card [2], which should be placed inside the nested loop, at the beginning of the answer field.
This suggests that the learner may have placed the card without fully understanding the code structure.

In this case, using the standard Levenshtein distance, simply inserting card [2] shortens the distance to the correct sequence [1][2][3][4][5][6], resulting in an evaluation that does not accurately reflect the learner's true understanding.
In contrast, the blank-considered distance treats the answer field as an array including blanks, such as [2][0][0][0][0], thus preventing the distance from shrinking merely by inserting a card, allowing inappropriate trials to be identified.
This tendency was also observed in the other learners, user36 and user74.
In other words, at the early stages of practice when the answer field contains few cards, the standard distance may overestimate the progress due to simplistic insertion operations, whereas considering blanks can prevent such overestimation.

Next, we focus on the changes after turn 8 of user36, leading up to the correct answer.
It can be seen that the extended distance considering dummy cards decreases more sharply than the standard distance.
As shown in Figure 8, user36 appears to have understood the structure of the nested loop from the early stages of the exercise.
In the latter stages, the learner approached the correct answer by replacing the condition statement for the inner for-loop with the correct card.
Such learning progress is difficult to capture using the standard distance, but the dummy card-corrected distance appropriately reflects operations that involve removing unnecessary cards, allowing the learning changes to be visualized more accurately.

Furthermore, in the case of user74, when viewed with the standard distance, there appears to be slight overall progress, suggesting some stagnation but a general movement toward the correct answer.
However, when viewed with the blank plus dummy-corrected distance, the distance sometimes increases up to 8, and overall, the distance barely decreases.
As shown in Figure 9, the necessary nested loop structure was not formed, and the ordering of the for-loops was inappropriate, indicating insufficient understanding of the program structure.
Thus, it is evident that the corrected distances serve as effective indicators that can highlight learning difficulties and insufficient understanding that are not captured by the standard distance.

5 Experiment

This chapter describes the experiments conducted to demonstrate the effectiveness of the proposed method. Specifically, learning logs from a programming exercise conducted in a university course are used, and the proposed extended Levenshtein distance is applied to analyze the learning process. The results of this analysis are presented.

6 Conclusion and Future Work

In this study, we aimed to refine the analysis of learning processes in constructive learning and proposed an extended method that incorporates the consideration of blanks and the correction for dummy cards into the Levenshtein distance. We then validated its effectiveness.

The results demonstrated that the proposed method could more accurately visualize inappropriate operations in the early stages of learning and the convergence process toward correct answers in the later stages, aspects that were difficult to capture with the standard distance metric. In particular, operations based on incorrect understanding and traces of trial and error were clearly reflected as fluctuations in the distance values, confirming that the proposed indicators are effective for quantitatively grasping the deepening of understanding and difficulties experienced by individual learners.

As future work, beyond time-series analysis of individual learning processes, it will be necessary to develop methods for aggregating data from multiple learners to visualize group trends. For example, it would be possible to extract features from the transition patterns of extended distances for each learner and classify them via clustering, thereby identifying different types of learning progression. Additionally, by visualizing the aggregated data using graphs or heatmaps, it is expected that improvements in overall lecture support and the identification of specific stumbling points can be achieved.

Furthermore, an important future research direction will be to establish a system that provides real-time feedback based on the extended distance analysis. This would enable learners to objectively monitor their own understanding and encourage appropriate self-corrective behaviors.

Based on the outcomes of this study, we plan to further develop our research toward broader applications in learning support.

Acknowledgement

This work was partly supported by Grant-in-Aid for Scientific Research (C) No. 22K02815 and No.23K02697 from the Japan Society for the Promotion of Science (JSPS).

References

[1] Papert, S. : “Constructionism: A New Opportunity for Elementary Science Education”, American Journal of Physics, Vol.49, No.3, pp.11–15 (1981).

[2] Natsumi Tanabe, Shimpei Matsumoto, “Analyzing the Learning Process in a Card Operation-Based Programming Learning Support System,” IIAI Letters on Informatics and Interdisciplinary Research, Vol. 5, DOI: 10.52731/liir.v005.314, September 2024.

[3] Papamitsiou, Z., & Economides, A. A. (2014). Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence. Journal of Educational Technology & Society, 17(4), 49–64.

[4] Boroujeni, M. K., & Dillenbourg, P. (2019). Diagnosing Learning Patterns and Behaviors through Temporal Analytics. International Journal of Learning Analytics, 6(2), 115–130.

[5] Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady, 10(8), 707–710.

[6] Wagner, R. A., & Fischer, M. J. (1974). The string-to-string correction problem. Journal of the ACM, 21(1), 168–173.

[7] Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3), 171–176.

[8] Shinpei Matsumoto, Yusuke Hayashi, Munetaka Hirasaki. "Development of a Programming Learning System Using Card Operation Focused on Considering Relationships Between Parts," IEEJ Transactions on Electronics, Information and Systems, Vol. 138, No. 8, pp. 999-1010 (2018).

[9] Ruka Murakami, Emiko Morinaga, Shinpei Matsumoto, Kengo Iwai, Yusuke Hayashi, Munetaka Hirashima. "A Basic Method Proposal for Analyzing the Learning Process of a Programming Learning System Using Card Operation," Proceedings of the 2018 Student Research Presentation of the Japan Society for Educational Technology, A14, pp. 181-182 (2019).