ABSTRACT

COMPUTATIONAL APPROACHES TO CONSTRUCT AND ASSESS KNOWLEDGE MAPS FOR STUDENT LEARNING

by Bao Wang

Knowledge maps have been widely used in knowledge elicitation and representation to eval- uate and guide students’ learning. To improve upon current computational approaches to construct and assess knowledge maps, this thesis adopts a hybrid methodology that combines machine learning techniques and network science. By providing methods to extract features to evaluate knowledge maps and expand the assessment scope by accounting for group inter- action and multiple expert maps, this thesis addresses the overall gap of current approaches for map construction and assessment. Specifically, this thesis offers three major contribu- tions: 1) identifying necessary and suﬀicient graph features for knowledge maps evaluation,

2) assessing the role of group interaction during knowledge map construction and how group size affects the quality of map construction, and 3) providing an algorithmic framework to capture differences between student maps and multiple expert maps. Finally, this thesis examines the implications for the fields of network science and educational technology of applying knowledge maps in student learning.

COMPUTATIONAL APPROACHES TO CONSTRUCT AND ASSESS KNOWLEDGE MAPS FOR STUDENT LEARNING

Thesis Submitted to the

Faculty of Miami University in partial fulfillment of

the requirements for the degree of Master of Science

by Bao Wang

Miami University Oxford, Ohio 2022

Advisor: Dr. Philippe J. Giabbanelli Reader: Dr. Michael A. Zmuda Reader: Dr. Khodakhast Bibak

©2022 Bao Wang

This thesis titled

COMPUTATIONAL APPROACHES TO CONSTRUCT AND ASSESS KNOWLEDGE MAPS FOR STUDENT LEARNING

by

Bao Wang

has been approved for publication by

The College of Engineering and Computing and

The Department of Computer Science & Software Engineering

Dr. Philippe J. Giabbanelli

Dr. Michael A. Zmuda

Dr. Khodakhast Bibak

## Contents

List of Tables [vi](#_bookmark0)

List of Figures [vii](#_bookmark1)

Dedication [ix](#_bookmark2)

Acknowledgments [x](#_bookmark3)

Quotation [xi](#_bookmark4)

1. Introduction [1](#_bookmark5)
   1. Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . [3](#_bookmark8)
      1. Aim One: Identify Necessary and Suﬀicient Conditions for Knowledge

Map Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . [3](#_bookmark10)

* + 1. Aim Two: Investigate the Role of Group Interaction in Knowledge

Map Construction . . . . . . . . . . . . . . . . . . . . . . . . . . . . [4](#_bookmark11)

* + 1. Aim Three: Develop an Algorithmic Framework to Capture the Dif- ferences between Student Maps and Multiple Reference Maps . . . . [4](#_bookmark12)
  1. Organization of the Thesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . [5](#_bookmark13)

1. Background [6](#_bookmark14)
   1. How to Elicit and Represent Knowledge as a Map? . . . . . . . . . . . . . . [7](#_bookmark15)
      1. Concept Map . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . [7](#_bookmark16)
      2. Causal Loop Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . [8](#_bookmark18)
      3. Causal Map [10](#_bookmark20)
   2. Why are Knowledge Maps Used in Assessment for Learning? [11](#_bookmark22)
      1. Types of Assessment for Learning [11](#_bookmark23)
      2. Criteria for Effective Formative Assessment [12](#_bookmark25)
      3. Reasons of Using Knowledge Maps in Formative Assessment [13](#_bookmark27)
   3. How to Assess Maps? [16](#_bookmark29)
      1. Properties of Knowledge Maps as an Assessment Tool [16](#_bookmark30)
      2. Reference-free Scoring Systems [17](#_bookmark31)
      3. Reference-based Scoring Systems [19](#_bookmark33)
      4. Application of Hybrid Systems [20](#_bookmark36)
   4. Open Problems in Assessing Maps: Text Mapping, Feedback, and Linguistic Variability [22](#_bookmark39)
      1. Converting Essays to Causal Structures: Text Mapping [22](#_bookmark40)
      2. Automated Feedback [24](#_bookmark42)
      3. Linguistic Variability [25](#_bookmark43)
   5. Summary [30](#_bookmark48)
2. Identifying Informative Features to Evaluate Student Knowledge as Maps [31](#_bookmark49)
   1. Introduction [32](#_bookmark50)
   2. Background [33](#_bookmark51)
      1. Core Concepts of Graph Theory in Map Assessments [33](#_bookmark52)
      2. Unsupervised Feature Selection Methods [35](#_bookmark54)
   3. Methods [39](#_bookmark59)
      1. Data Collection [39](#_bookmark60)
      2. Data Pre-processing [40](#_bookmark62)
      3. Data Analysis [41](#_bookmark64)
   4. Results [43](#_bookmark67)
   5. Discussion [46](#_bookmark71)
   6. Conclusion [47](#_bookmark72)
3. Transitioning from Individuals to Groups in Knowledge Map Construction [48](#_bookmark73)
   1. Introduction and Literature Review [49](#_bookmark74)
   2. Research Questions [50](#_bookmark76)
   3. Methods [50](#_bookmark77)
      1. Data Collection [50](#_bookmark78)
      2. Data Pre-processing [51](#_bookmark79)
      3. Data Analysis [52](#_bookmark81)
   4. Results [55](#_bookmark86)
      1. Q1: Transitioning from Individuals to Group Maps [55](#_bookmark87)
      2. Q2: Potential mediating effect of the group size [59](#_bookmark92)
   5. Discussion [61](#_bookmark94)
   6. Conclusion [62](#_bookmark95)
4. Algorithmic Approaches to Guide a Student’s Map Using Multiple Expert Maps [64](#_bookmark96)
   1. Introduction [64](#_bookmark97)
   2. Reference-based Methods for Map Comparison [66](#_bookmark99)
      1. Global Graph Metrics [66](#_bookmark100)
      2. Adjacency Matrices [67](#_bookmark101)
      3. Graph Edit Distance [67](#_bookmark102)
      4. Graph Embedding [68](#_bookmark104)
      5. Tversky Similarity [68](#_bookmark105)
   3. Framework [69](#_bookmark107)
      1. Main Elements in the Framework [69](#_bookmark108)
      2. Four Distinct Comparison Strategies [70](#_bookmark110)
   4. Discussion and Future Work [75](#_bookmark116)
      1. Analysis of Four Comparison Strategies [75](#_bookmark117)
      2. Future Work [76](#_bookmark118)
   5. Conclusion [77](#_bookmark119)
5. Designing the Next Generation of Map Assessment Systems [78](#_bookmark120)
   1. Introduction [79](#_bookmark121)
   2. First Task: Finding Suﬀicient Criteria to Evaluate the Structural Quality of

a Learner’s Map [81](#_bookmark123)

* 1. Second Task: Guiding a Learner in Progressing Towards a Multiplicity of Experts [83](#_bookmark126)
  2. Third Task: Improving Software Usability via a Practical Design Framework [86](#_bookmark128)
  3. Discussion [87](#_bookmark130)

1. Conclusion [89](#_bookmark131)
   1. Contributions [89](#_bookmark132)
   2. Limitations and Future work [90](#_bookmark133)

References [91](#_bookmark133)

## List of Tables

* 1. An overview of research on criteria for effective formative assessment. [13](#_bookmark26)
  2. Knowledge maps meet the criteria for effective formative assessment. [15](#_bookmark28)
  3. A summary of different ontology matching systems. [29](#_bookmark47)
  4. A list of metrics that have been used in studies of knowledge maps [34](#_bookmark53)
  5. Metrics for data analysis along with how they are implemented. [41](#_bookmark65)
  6. Eight selected UFS algorithms and what they output, along with their imple- mentations. [42](#_bookmark66)
  7. Metrics distribution across all five data sets of collected maps. [43](#_bookmark68)
  8. Sum of rank and average rank of each feature in the result. [45](#_bookmark70)
  9. Descriptions and implementation of each graph metric. [53](#_bookmark82)
  10. A sample of how maps and their graph metrics are documented in the spread- sheet. [54](#_bookmark84)
  11. Descriptive statistics of graph metrics in maps. [56](#_bookmark88)
  12. Results of the one-way ANOVA test. [58](#_bookmark90)
  13. Results of Tukey’s HSD Test [60](#_bookmark93)
  14. A list of software for map comparison. [65](#_bookmark98)
  15. Main elements of the framework. [70](#_bookmark109)

## List of Figures

* 1. Three common types of knowledge maps. . . . . . . . . . . . . . . . . . . . . [1](#_bookmark6)
  2. An example of automated feedback function in ITACM. . . . . . . . . . . . . [2](#_bookmark7)
  3. Three aims of this study and broader contributions in academia. . . . . . . . [3](#_bookmark9)
  4. Concept map of concept maps. . . . . . . . . . . . . . . . . . . . . . . . . . [8](#_bookmark17)
  5. Simple causal loop diagram with time delay from O to P. . . . . . . . . . . . [9](#_bookmark19)
  6. An example of causal map. [10](#_bookmark21)
  7. A formative assessment process cycle. [12](#_bookmark24)
  8. A graphical illustration of the relation between two properties sets. [20](#_bookmark35)
  9. Transformation of raw data from a model *M* to a standardized format. [21](#_bookmark37)
  10. Model *M* and reference model *Mr*, and their spanning trees with two most distant nodes marked. [22](#_bookmark38)
  11. Reference and student map generated by GIKS. [24](#_bookmark41)
  12. Implementation of ITACM in generating feedback. [26](#_bookmark44)
  13. An example of linguistic variability in student and expert maps. [27](#_bookmark45)
  14. Matching techniques classification. [29](#_bookmark46)
  15. An example of unsupervised feature selection approach. [35](#_bookmark55)
  16. The Framework of filter feature selection methods. [36](#_bookmark56)
  17. The Framework of wrapper feature selection methods. [37](#_bookmark57)
  18. Sample expert map from Dataset A [38](#_bookmark58)
  19. Sample student map from Dataset B [39](#_bookmark61)
  20. Overview of the data cleaning process. [40](#_bookmark63)
  21. Results of feature rank/selection from eight different UFS algorithms. [44](#_bookmark69)
  22. A learning cycle presents how students carried out a case study via the problem-based approach. [49](#_bookmark75)
  23. An overview of data cleaning process. [51](#_bookmark80)
  24. An example of how a map is evaluated via graph metrics. [54](#_bookmark83)
  25. Similarity between collaborative map of group 4 and the maps previously produced by its individuals. [55](#_bookmark85)
  26. Distribution of graph metric values in both individual work and group work. [57](#_bookmark89)
  27. Similarity analysis of the composition for each group map. [59](#_bookmark91)
  28. An example of GED. [68](#_bookmark103)
  29. An example of graph embedding in 3D space. [69](#_bookmark106)
  30. Detailed process and an example of *Strategy One* [71](#_bookmark111)
  31. Detailed process and an example of *Strategy Two* [72](#_bookmark112)
  32. Detailed process and an example of *Strategy Three* [73](#_bookmark113)
  33. Different paths with information of total edits and the change of similarity to

each expert map in each path. [73](#_bookmark114)

* 1. Detailed process and an example of *Strategy Four* [74](#_bookmark115)
  2. The assessment of learners’ knowledge externalized in the form of causal maps. [79](#_bookmark122)
  3. An example of a map assessed via graph-level metrics. [82](#_bookmark124)
  4. Process to identify the most important metrics via feature extraction and feature selection. [83](#_bookmark125)
  5. One way to handling the multiplicity of expert maps. [85](#_bookmark127)
  6. Human-Computer Interaction as informing design principles for learning. [87](#_bookmark129)

# Dedication

This thesis is dedicated to memory of my dad, 王朝秋. Thank you for bringing me lots of love, laugh, and happiness in my childhood.

# Acknowledgments

First and foremost, I would like to express my sincere gratitude to Dr. Philippe J. Giabbanelli, who has been mentoring me, encouraging me, inspiring me, and teaching me the priceless wisdom of conducting research as a scholar. Thank you for the untiring support for the research project, in particular when I am in China. I would not have been able to pull off the extremely challenging task of completing my thesis project and writing it up in a foreign language without your unwavering support.

Next, I would like to thank my thesis committee, Dr. Michael A. Zmuda and Dr. Kho- dakhast Bibak, for your willingness to support my work and all the detailed suggestions that you patiently offered me. I would also like to thank you for encouraging me for pursuing this CS M.S. degree.

In addition, I am deeply thankful to my professors from the Department of Computer Science and Software Engineering at Miami University, whose excellent teaching cultivated me as a computer scientist: Drs. John Femiani, Daniela Inclezan, Jim Kiper, Eric Rapos, DJ Rao, and Mr. Michael Stahr.

Finally, I thank my beloved family. I am forever indebted to my mom and elder sister,赖财兰 and 王丽文, who have always been my role models in life. Thank you for your unconditional love. I am immensely grateful to my wife, 张婧, for trusting me and assuring me that I could achieve the degree. I also thank my parents-in-law, 阎杰 and 张云生, for believing in me and supplying me with infinite love and care. I also thank my dearest

family/friends from the U. S. for their support: the Whalens, Richelle Frabotta, Paige Kostoff, and the Evans.

# Quotation

Stay hungry. Stay foolish. — *Whole Earth Catalog* 1180, October 1974. Quoted by Steven Paul Jobs (‘Steve Jobs’) in the Commencement address delivered at Stanford University on June 12, 2005.

**Chapter 1**

# Introduction

Since 1972, knowledge maps (Figure [1.1](#_bookmark6)) have been used widely as a learning, pedagogical, and assessment tool in the field of education for supporting students’ meaningful learning due to their ability to represent and visualize learners’ internal knowledge structures [[1](#_bookmark134), [2](#_bookmark135)]. Through knowledge maps, learners are able to elicit and externalize their organization of knowledge structures, which provides direct feedback for instructors to assess learners’ understanding of content and offers useful information for learners to bridge their knowledge gaps [[3](#_bookmark136), [4](#_bookmark137)].

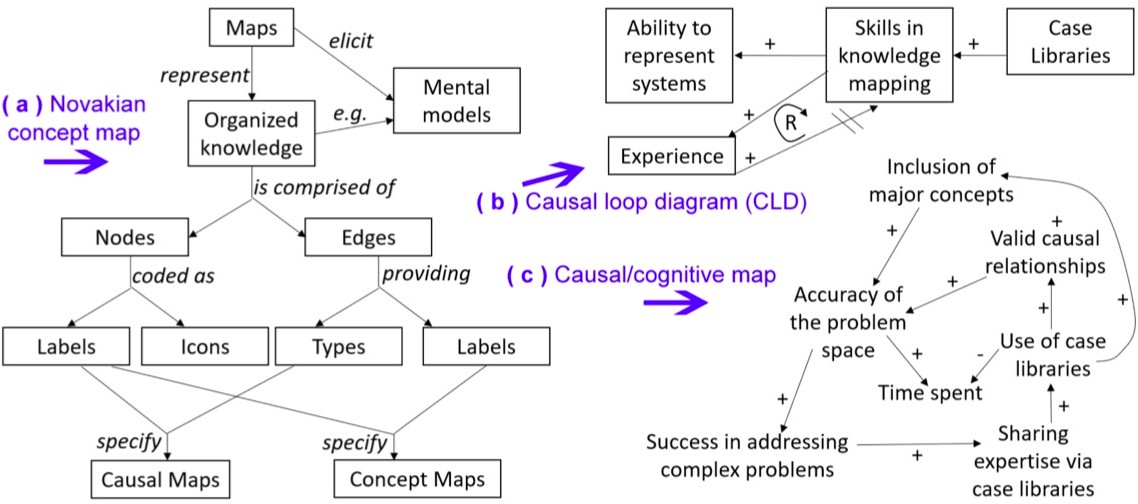


Figure 1.1: Three common types of knowledge maps. Reproduced from Giabbanelli and Tawfik [[5](#_bookmark138)]. Details of the above maps are explained in Section [2.1](#_bookmark15).

Thanks to the rapid development of computer technology in the past decades, the appli- cation of computer-based mapping techniques and systems in education has improved both in terms of methodology and in empirical findings. Research has shown that students gain better understanding of knowledge and benefit more in learning improvement via computer- aided approaches than those who learn with traditional approaches [[1](#_bookmark134), [6](#_bookmark139)]. In addition, the computer-based knowledge maps techniques and systems ease the process of map construc- tion (e.g, SMD [[7](#_bookmark140)], GIKS [[8](#_bookmark141)]), advance the interaction and collaboration among teachers and students (e.g, CRESST [[9](#_bookmark142)]), and provide automated feedback to students to improve their

learning (e.g, ITACM [[3](#_bookmark136)] in Figure [1.2](#_bookmark7)).

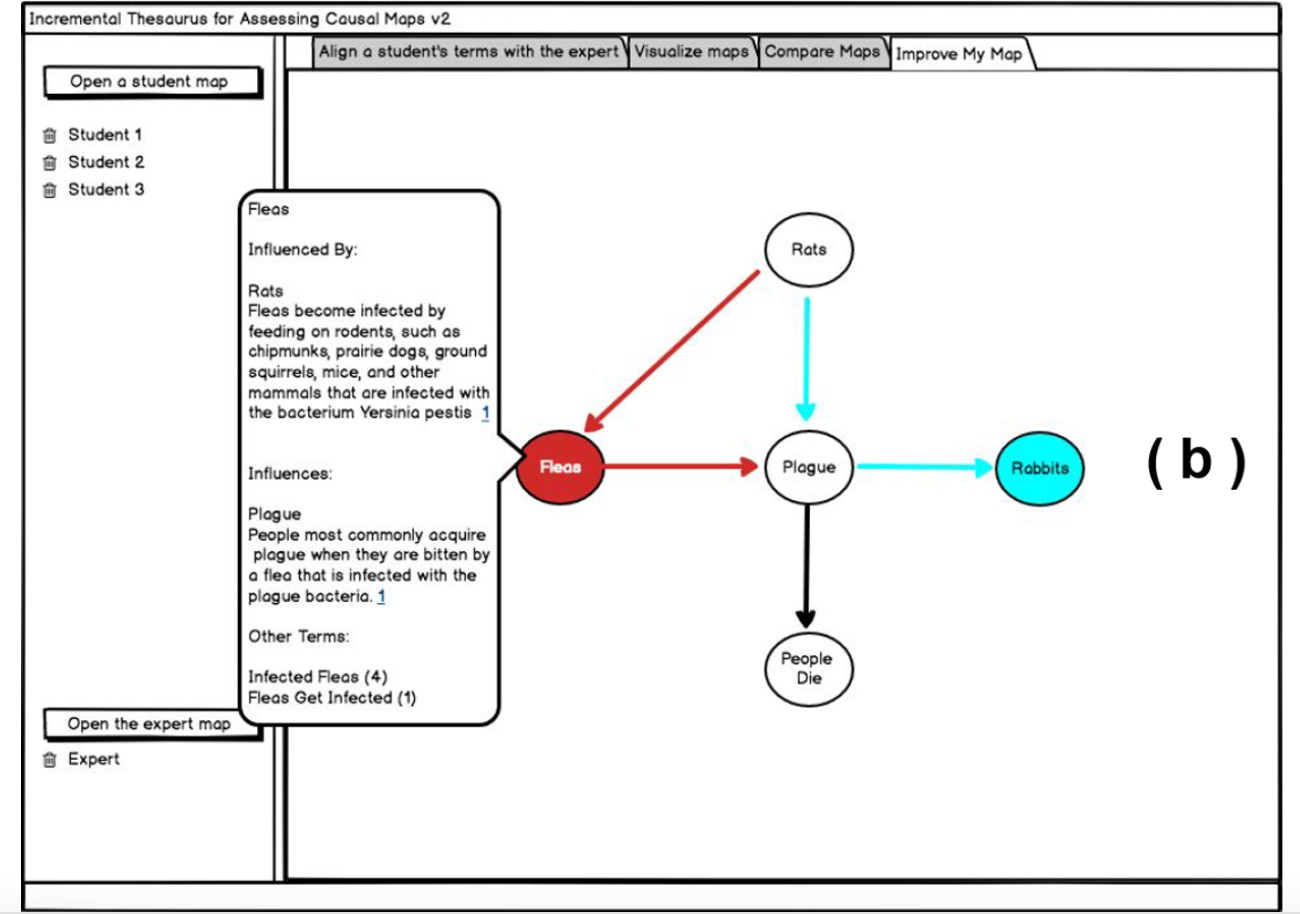


Figure 1.2: An example of automated feedback function in ITACM. Differences between the student and the expert’s map (e.g., missing nodes in red that present only in the expert’s map and extraneous nodes in blue that present only in the student’s map) are illustrated with contextual explanation. Reproduced from Giabbanelli and Tawfik [[3](#_bookmark136)].

Although learning with knowledge maps appears promising, several obstacles remain when constructing and evaluating knowledge maps, among which the following are central to this thesis:

* Inconsistent assessment metrics in assessing the construct of maps. For example, some studies may choose one set of graph metrics (e.g., features including the diameter of spanning tree) while others choose another set of metrics (e.g., features excluding the diameter of spanning tree).
* Missing explanation of group interaction in map construction. Studies of map as- sessment usually focus solely on the final product of map construct to evaluate the similarities and differences between individual maps and expert maps while neglecting the potential influence of group interaction and group size.
* A lack of consideration of diverse assessment standing points. One common approach

of map assessment is to compare students’ maps with only one expert map and evaluate the learning performance based on the similarities between both maps. However, this approach is insuﬀicient to account for questions such as “do different viewpoints that result in maps with vast differences promote students’ learning?”

In summary, the overall objective of this study is to address the three obstacles above to improve the assessment of students’ understanding of knowledge via graph structures. In addition, the broader impacts of this study are an improvement in the consistency of assessment and a broadening of the scope that takes into account groups or multiple solution paths to open-ended problems.

## Contributions

This study seeks to investigate multiple computational approaches of constructing and assessing knowledge maps for student learning. Specifically, the contributions are reflected in the following three complementary aims (Figure [1.3](#_bookmark9)).

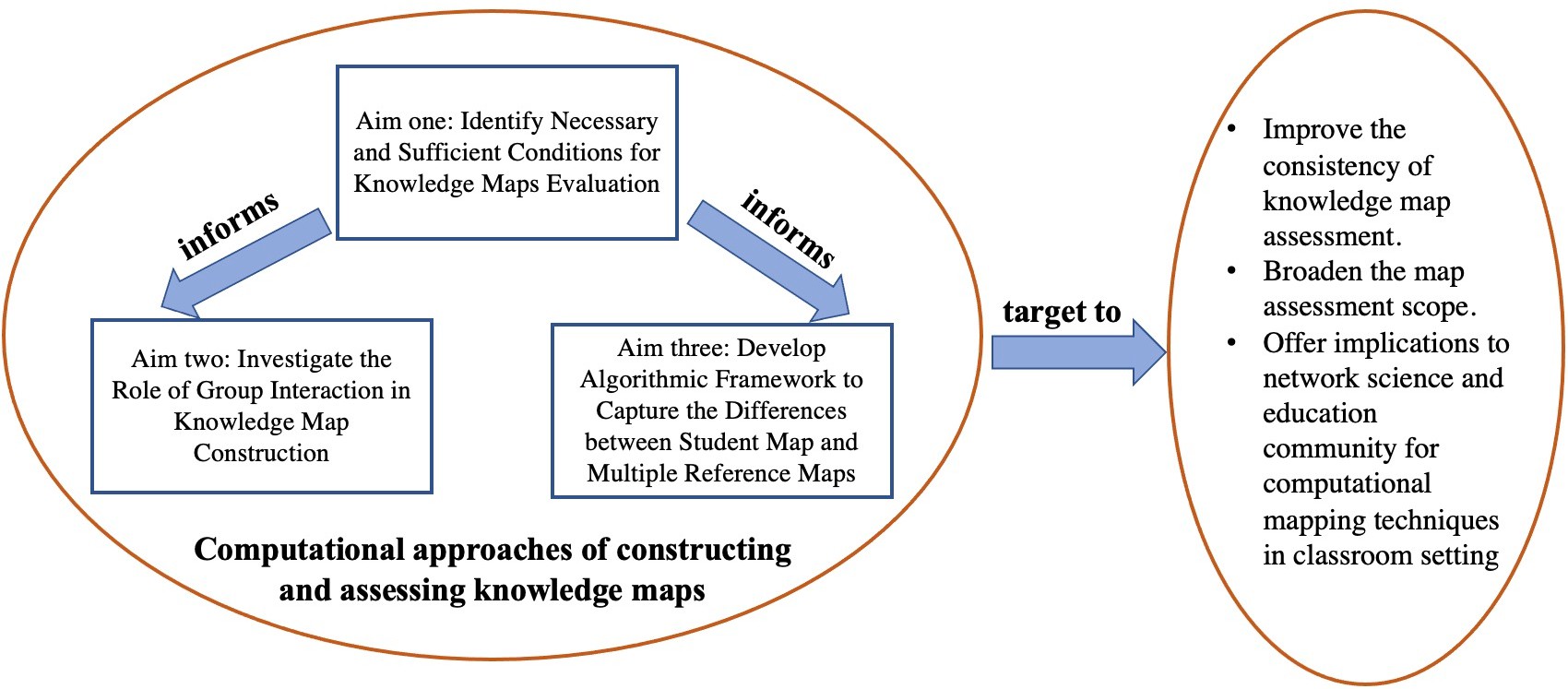


Figure 1.3: Three aims of this study and broader contributions in academia.

### Aim One: Identify Necessary and Suﬀicient Conditions for Knowledge Map Evaluation

In practice, there are two common methods to evaluate knowledge maps: reference- free and reference-based assessment, which are discussed in Section [2.3.2](#_bookmark31) and Section [2.3.3](#_bookmark33) respectively. In the reference-free assessment, knowledge maps are often considered as graphs and graph theory is thus frequently applied to assess maps [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [10](#_bookmark143), [11](#_bookmark144)]. Graph metrics, such as number of nodes, number of edges, diameter, total number and average lengths of

cycles, total number and longest length of chains, are considered as important measurements in map assessment.

However, current studies that use reference-free assessment usually count on different measurement metrics to evaluate knowledge maps based on research topics and knowledge domains. Each scholar has his or her preference for selecting their list of graph features for evaluation. Few studies discuss what conditions actually correlate with the expected quality of map construction, or what set of measures truly reflects knowledge structures of learning. To address this gap, We adopt machining learning techniques (e.g., feature selection) to investigate necessary and suﬀicient conditions in evaluating knowledge maps via analyzing 202 maps originating from four case studies. Detailed procedures and results are shown in Chapter [3](#_bookmark49).

### Aim Two: Investigate the Role of Group Interaction in Knowledge Map Construction

Knowledge maps, as structured visual representations of domain knowledge, have been widely incorporated into pedagogical and assessing tools for decades [[1](#_bookmark134)]. Although knowledge maps can be utilized in both individual and collaborative learning activities to promote students’ learning [[2](#_bookmark135)], research rarely examines the influence of group interaction on map construction in knowledge representation and what size of group will possibly lead to good “group thinking” that produces stronger knowledge representation. Therefore, one aim of this study is to investigate the impact of group interaction by comparing map structures generated from individuals and those from groups of different sizes. Through analyzing such differences by computing different graph metrics (e.g., number of nodes, number of edges, diameter of graph), nuances of map quality corresponding to different group sizes could be identified. This innovative use of network science in relation to group size has practical implications for teachers, for example by identifying whether there are benefits to having a larger group up to a given size. In Chapter [4](#_bookmark73), we examined how 12 graph metrics (e.g., number of nodes, diameter) change when we transition from the individual maps of learners to the maps produced by groups.

### Aim Three: Develop an Algorithmic Framework to Capture the Differences between Student Maps and Multiple Refer- ence Maps

As stated in Section [1.1.1](#_bookmark10), reference-based assessment is another frequently utilized method in evaluating knowledge maps [[3](#_bookmark136), [4](#_bookmark137), [7](#_bookmark140), [12](#_bookmark145)]. By comparing a map with a refer- ence/expert map, we may identify the knowledge gap between students and experts. Thus, instructors are able to provide corresponding feedback based on the similarities and differ- ences of the maps to facilitate students’ learning [[4](#_bookmark137)]. However, studies that use reference- based assessment usually compare students’ maps with only one expert map, which implicitly assumes that there is only one standard answer and students should think in only one specific

way. In reality, this is not always effective when improving students’ learning, especially for open-ended questions or ill-structured problems, which often admit a much broader set of perspectives and are often contextualized [[13](#_bookmark146)]. For example, for the open-ended question: should guns be banned in the United States? Different experts may hold various viewpoints. Thus, it would result in different map structures. If students’ maps are evaluated based on one single expert map and students’ learning is promoted by one single perspective, students would lost the opportunities to be exposed to more comprehensive views and develop holis- tic cognitive thinking. Furthermore, research regarding human learning shows that multiple sources of information (e.g., legitimate/ questionable/ opposite information) are more ben- eficial to develop critical thinking and improve knowledge understanding [[14](#_bookmark147)][pp. 420-421]. In particular, research in *case library* show that information provided by experts could shape the knowledge representation of students [[15](#_bookmark148)]. Thus, it is essential to consider and integrate multiple perspectives into the assessment systems of knowledge maps. In this study, we develop an algorithmic framework including four strategies for map comparison to capture the differences between student maps and multiple expert maps that provide useful feedback to improve students’ learning. Detailed explanations are presented in Chapter [5](#_bookmark96).

## Organization of the Thesis

The thesis is organized as follows. In Chapter [2](#_bookmark14), we provide background information of knowledge maps including various types of knowledge maps, reasons for applying knowledge map as a learning tool, and various methods to analyze and evaluate knowledge maps, as well as discuss open problems that affect the application of knowledge maps and mapping techniques in student learning. Then, in Chapter [3](#_bookmark49), [4](#_bookmark73), and [5](#_bookmark96), we present our research designs and findings to achieve the above three aims. In Chapter [6](#_bookmark120), we revisit the open questions and opportunities to automatically assess a student’s knowledge as a map in designing the next generation of map assessment systems. Lastly, a brief conclusion will be provided to highlight the key points of this thesis in Chapter [7](#_bookmark131).

**Chapter 2**

# Background

Based on Ormrod’s definition of learning “as a long-term change in mental representa- tions or associations as a result of experience” [[14](#_bookmark147)], the learning entity *resides implicitly in a person’s mind*, which makes it impossible to assess learning *directly*. Thus, knowledge elic- itation and representation become essential when it comes to evaluating and guiding either the learning *process* (i.e., formative assessment) or the learning *outcomes* (i.e., summative as- sessment) of learners. The knowledge externalization can be realized through several ways, such as speaking out aloud, writing a text, drawing a picture, or constructing knowledge maps [[7](#_bookmark140)]. Various graph-based conceptual models, including Novakian concept maps, causal loop diagram, causal map, and rich pictures, are frequently applied in eliciting, representing, and assessing knowledge learning [[1](#_bookmark134), [16](#_bookmark149), [17](#_bookmark150)]. Specifically, the concept of knowledge mapping is widely used in formative and summative assessment of students’ learning in the field of education [[1](#_bookmark134), [4](#_bookmark137), [8](#_bookmark141)].

However, to guarantee the effective and eﬀicient use of knowledge maps in assessment, several aspects must be carefully designed and implemented. These include the *reliability* of using knowledge maps, effective methods to *assess* maps, *integration* of knowledge maps with other practices (e.g., by converting essays into maps), generating *guidance* (i.e., pro- viding feedback to learners), and addressing challenges caused by the *linguistic variability* of terminologies used in map construction.

Therefore, to lay down the groundwork of this thesis, I first summarize various ways of externalizing knowledge as a graph along with examples in Section [2.1](#_bookmark15), including Novakian concept map, causal loop diagram, and causal map. Then, I demonstrate in Section [2.2](#_bookmark22) why knowledge map is an effective tool for assessing learning. This section accounts for differ- ent types of assessment and various criteria to achieve effective formative assessment, while providing supportive evidence of using knowledge maps in formative assessment. Next, Section [2.3](#_bookmark29) presents theories and methods of evaluating knowledge maps, which includes reference-free assessment, reference-based assessment, and the real-world applications. In Section [2.4](#_bookmark39), I closely examine the open problems that affect the application of knowledge maps in formative assessment in practice and current attempts at resolving those issues. Fi- nally, a brief summary is presented to highlight the key points of this background knowledge research.

## How to Elicit and Represent Knowledge as a Map?

This section provides an overview of three most commonly used forms of knowledge maps: Novakian concept map, causal loop diagram, and causal map. Concepts, characteristics, and general ways to generate the three knowledge maps above are demonstrated in detail along with corresponding figures. To remain consistent, this paper adopts knowledge maps as an umbrella term to describe all graph-based conceptual models that are discussed in this chapter.

### Concept Map

The notion of *concept map* was first created and developed by Joseph Novak and his team in 1972, who aimed at understanding changes in children’s learning of science [[18](#_bookmark151)]. As shown in Figure [2.1](#_bookmark17), a concept map is a graphical tool to illustrate potential relation- ships between various concepts as a graph. It is generally hierarchically structured, with a focus question at the head (e.g., the notion of Concept Maps at the top of Figure [2.1](#_bookmark17)) guiding the generation of ideas or questions that the concept map attempts to explicate and solve. The map incorporates various concepts as nodes, which are labeled with words and/or symbols. Concepts, as nodes, are connected via links with propositions that identify the in- terrelationship or dependency of two nodes; those propositions are simple, meaningful, and easy to understand. The propositions are formed in the structure of concept-link-concept “triple” [[17](#_bookmark150)]. The arrowheads in the link indicate the flow direction within the concept map.

A general approach to create a a concept map has been described and discussed as follows [[2](#_bookmark135), [17](#_bookmark150), [19](#_bookmark152)]:

* + - 1. ***Define a focus question***: a focus question within a domain specifically identifies the issue that the concept tries to address, guides idea generation, and leads the direction of mapping.
      2. ***Identify key concepts***: the next step is to identify key concepts regarding the identified domain. The notion of concept is defined as “a perceived regularity or pattern in event or objected designated by a label” [[17](#_bookmark150)].
      3. ***Arrange concepts by inclusiveness or priority***: after the key concepts are identi- fied, those concepts will be ordered and spatially arranged based on their inclusiveness or priority. The broadest concepts are set on the top of the list while more specific concepts at the bottom. The spatial arrangement of key concepts is called “parking lot” by Novak and Cañas [[2](#_bookmark135)]. This may be accomplished by positioning sticky notes on a board.
      4. ***Create links***: When all concepts are set into the nodes, links with arrows to connect the concepts will be generated to indicate the relationships between concepts. The links are labeled with words or symbols to specify the detailed relationship.
      5. ***Iterate and revise spatial arrangement***: the process of arranging and linking concepts is iterative, hence the process can cycle back to step 2. Learners are expected to revisit and reexamine the accuracy of the relationship and several revisions are expected to guarantee the holistic representation of domain knowledge.

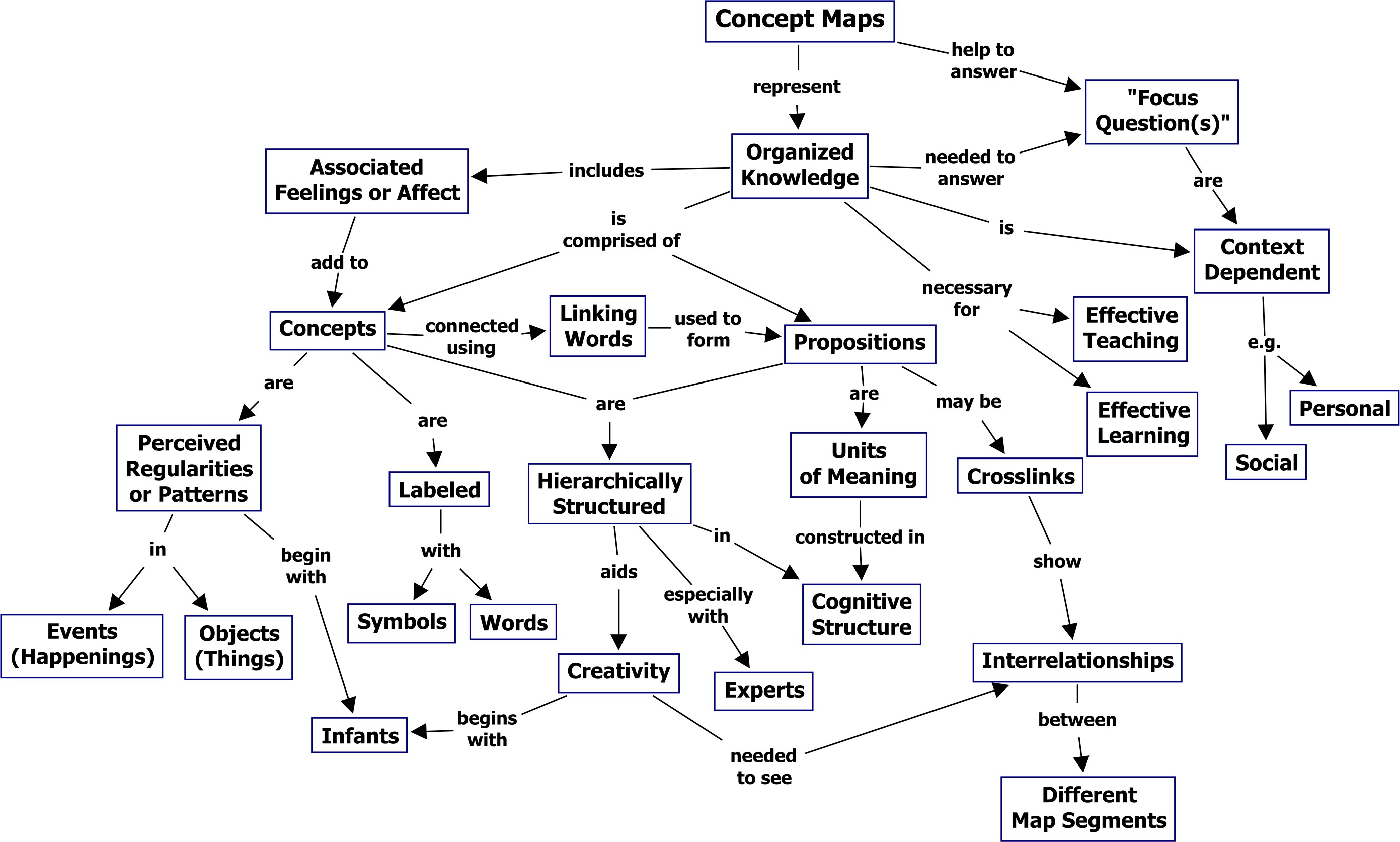


Figure 2.1: Concept map of concept maps. Reproduced from Novak and Cañas [[2](#_bookmark135)].

### Causal Loop Diagram

Another natural way to elicit and represent knowledge is to denote cause-effect relation- ships and characterize those relationships to understand the behaviors in the system. Causal loop diagrams is one of the most convenient ways to represent the feedback loop structure of systems [[20](#_bookmark153)]. Causal loop diagrams are often applied to illustrate the feedback loop struc- ture before creating a simulation model known as System Dynamics (SD). The feedback loop structure consists of two or more variables with cause-effect relationships denoted among the variables. The relationship of variables can be positive or negative and it is denoted with a

+ or − sign along with the arrow line.

As shown in Figure [2.2](#_bookmark19), a simple causal loop diagram consists of two feedback loop

structures, which are O + P + O and P + Q - P respectively. In the feedback loop O + P + O,

−→

−→

−→

−→

−→

−→

the arrow line from *O* to *P* with a + sign means a *positive* cause-effect relationship, where

an increase of *O* will cause an *increase* of *P* . While in the feedback loop P + Q - P, the arrow

−→

−→

line from *Q* to *P* with a − sign means a *negative* cause-effect relationship, where an increase of *Q* will cause a *decrease* of *P* . If the total number of negative relationships in a single feedback loop is even, then the loop is considered as positive (i.e., reinforcing), and the loop

is denoted as *R*+; if the the total number of negative relationships in a single feedback loop is odd, then the loop is considered as negative (i.e., balancing), and the loop is denoted as *B*− [[20](#_bookmark153), [21](#_bookmark154)]. The rationale for this categorization is that effects *multiply*: the product of an

even number of − signs is a + sign.

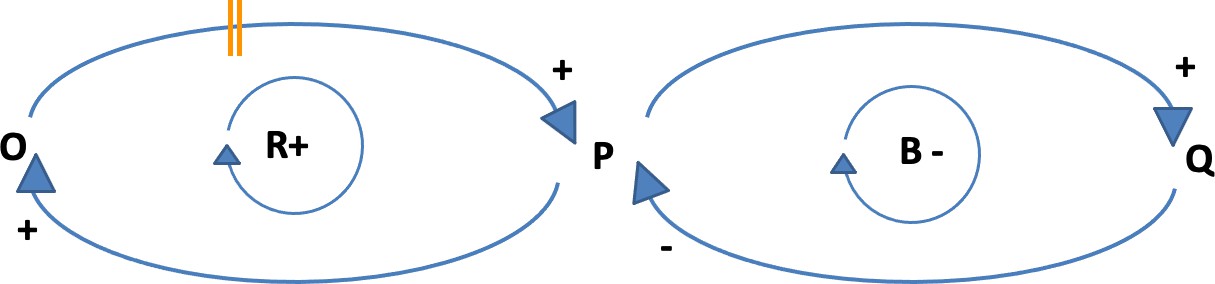


Figure 2.2: Simple causal loop diagram with time delay (i.e., two hash makes) from O to P.

Bala, Arshad, and Noh describe the general procedures to develop a causal loop diagram as follows [[20](#_bookmark153)], where the first two steps with the process for concept maps are similar:

* + - 1. ***Define the problem and objectives***. It is essential to study the system based on information collected from interviews, reports, and case studies to decide on the behavior mode of the system.
      2. ***Identify key elements*** that affect the behavior of the system. These key variables become the starting points in constructing the whole causal loop diagram.
      3. ***Identify secondary important elements***. The secondary layers of variables that connect and affect the key variables should be identified and added into the causal loop diagram.
      4. ***Identify tertiary important elements***. The tertiary layers of variables that con- nect and affect the secondary variables should be identified and added into the causal loop diagram. Tertiary variables with little importance can be omitted when con- structing the causal loop diagram. Although the process of Bala, Arshad, and Noh stops at tertiary elements, practitioners may continue to create more layers as long as variables remain within the scope of the model.
      5. ***Define cause-effect relationships*** and link the variables with arrows, following the order of key variables first, then secondary variables, and lastly tertiary variables.
      6. ***Identify closed loops***. One important characteristic is the closed loop in the causal loop diagram, so it is important to trace and identify the closed loop that describes the system.
      7. ***Identify balancing and reinforcing loops***. After identifying the closed loops, count the number of negative cause-effect relationships in each closed loop. Those with an odd number are considered as balancing loops, while the others are reinforcing loops.

Although the procedure above does not utilize this feature, causal loop diagrams can also represent time delays [[22](#_bookmark155)]. This is a useful feature, since the extension of these maps into a System Dynamics model will take temporal effects into account. Notice in the Figure [2.2](#_bookmark19) that there are two hash marks || , on the causal link between O and P, which represents a time delay. It means that some time is needed before an influence takes effect (i.e., it takes

some time for O to have a positive effect on P).

### Causal Map

A causal map (also known as a “cognitive map”) is another common representation of knowledge models, which has been widely applied in studies of managerial cognition, strategy, and decision-making [[23](#_bookmark156), [24](#_bookmark157)]. A causal map is also frequently accepted and applied in analyzing causal relationships [[24](#_bookmark157)], and it is widely used to elicit expert knowledge of certain domains to represent causal relationships of related factors that describe or explain a specific complex problem [[23](#_bookmark156), [25](#_bookmark158)].

As shown in Figure [2.3](#_bookmark21), a causal map consists of various factors/concepts with links between them. Those factors/concepts are generated from expert knowledge and the links with signs of + or − indicate positive or negative causal relationships of factors/concepts. The main purpose of a causal map is *not* to explicate why or how factors or concepts have causal relationships, but to explore the pattern or structure of their causal relationships [[23](#_bookmark156)].

In return, the patterns or structures obtained from the causal map facilitate the process of problem-solving and decision-making for a complex problem. In contrast with causal loop diagrams, a causal map does *not* represent time (e.g., edges are not coded to identify delays) and they are *not* annotated to indicate loops.

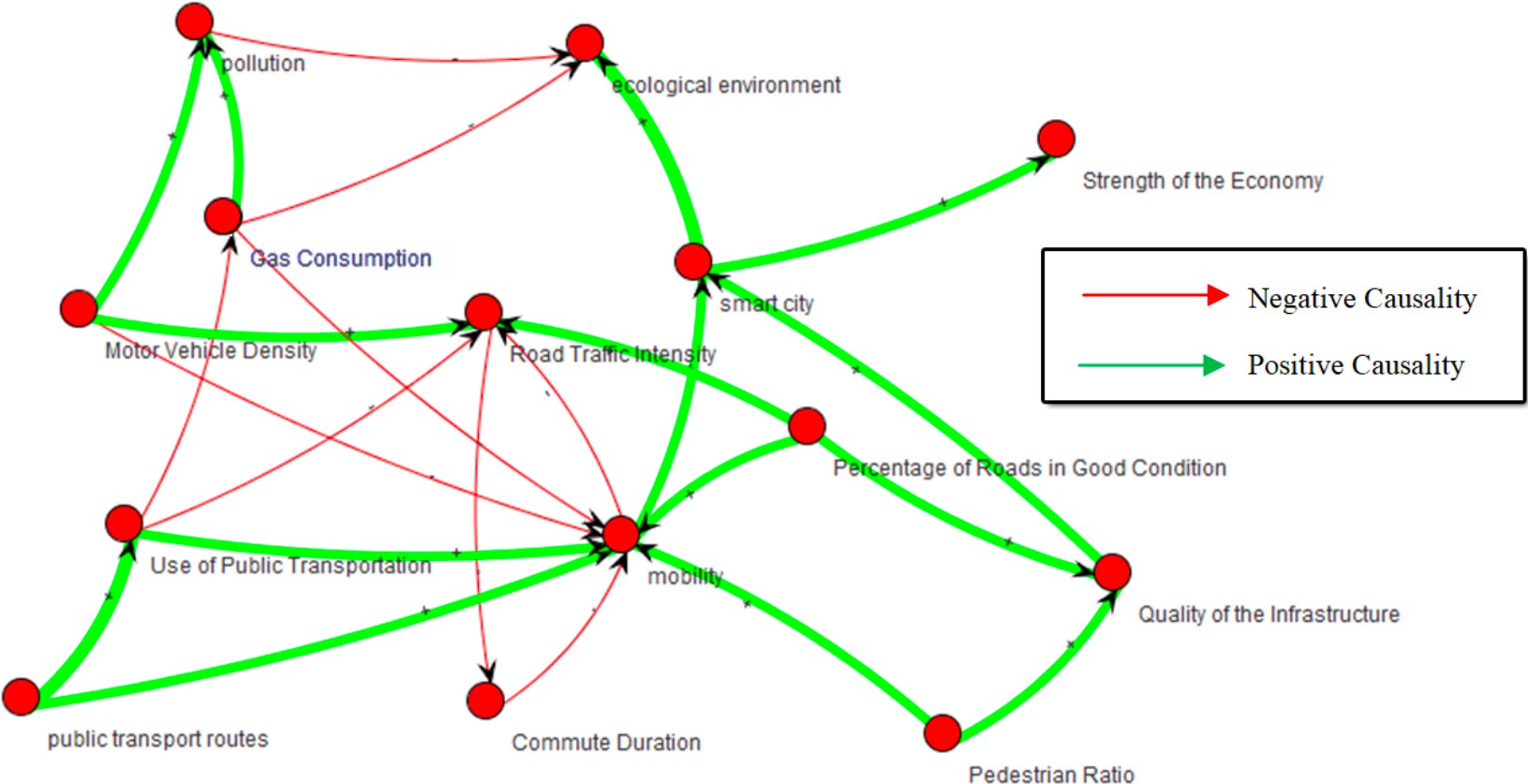


Figure 2.3: A causal map. Reproduced from Firmansyah et al. [[25](#_bookmark158)].

Nelson et al. suggest a three-step procedure to generate a causal map for an identified problem as follows [[24](#_bookmark157)]:

* + - 1. ***Data elicitation***: the first step is to decide on relative factors regarding one specific domain. Data could be collected through interviews, conversations, or collaboration with experts in that domain.
      2. ***Construct causal maps***: the second step is to identify causal connections and build raw individual causal maps. These connections may be identified from transcribed interviews, text mining of large corpora (e.g., Tweets), or reports. Individual maps then will be reviewed, re-casted, and aggregated based on the conceptually relevant concepts in individual maps.
      3. ***Validation of causal maps***: after the draft of aggregated map is constructed in step two, modelers consult experts to test the interpretive accuracy of concept nodes and links. This validation can be done structurally (e.g., analyzing the viability of a few paths in the map) or by extending the map into simulations, using techniques such as Fuzzy Cognitive Mapping. In a fuzzy cognitive map, the relations between the elements (e.g., concepts) of a “mental landscape” can be used to compute the weight of influence on those elements [[26](#_bookmark159)].

## Why are Knowledge Maps Used in Assessment for Learning?

Section [2.2](#_bookmark22) focuses on three parts: 1) two different types of assessments and their appli- cations, 2) a general introduction to criteria that have been considered to achieve an effective formative assessment, and 3) exploration of why knowledge maps and mapping techniques are reliable and feasible methods in formative assessment.

### Types of Assessment for Learning

Based on the National Research Council (NRC) regulations, education assessment has been categorized into two types: summative and formative assessments [[27](#_bookmark160)]. Summative assessment, such as standardized tests and exams, is designed to summarize the learning outcomes and evaluate the end product of learning, while formative assessment is an on-going process, which is designed to understand the learning process and facilitate learning [[1](#_bookmark134), [28](#_bookmark161), [29](#_bookmark162)]. The main distinction between summative and formative assessment is that the former is the assessment *of* learning while the latter is the assessment *for* learning.

Summative assessment has been used routinely in all aspects in assessing students’ learn- ing in education from elementary school to higher education. Students’ learning performance is assessed based on certain fixed criteria that students are expected to meet. As NRC states, one of the assessment purposes is to classify students’ performance by comparing them with each other [[27](#_bookmark160)]. Therefore, it is not uncommon to see that summative assessment is used at all levels of education to prove students’ accreditation of academic programs and to evaluate students’ internship and employment.

Formative assessment focuses more on the collaborative learning process between edu- cators and students [[1](#_bookmark134), [30](#_bookmark163)]. The purpose of formative assessment is to deepen students’ understanding and help them improve through identifying their strengths and weaknesses, providing feedback, and facilitating modification of learning activities [[1](#_bookmark134)]. *Feedback and modification* are critical to formative learning, and they are the key parts that distinguish

formative assessment from summative assessment [[28](#_bookmark161)]. As Cook summarizes, formative as- sessment could help students identify areas of strength and areas needing improvement; meanwhile, it provides instructors with feedback on how students have mastered domain knowledge, thus helping instructors to further facilitate students’ learning and to achieve learning objectives [[28](#_bookmark161)].

Formative assessment is an on-going process that involves states of before, during, and after the assessment and that aims to enhance students’ learning. It is a process that calls for attention throughout the whole learning process. A formative assessment process cycle is delineated in the Figure [2.4](#_bookmark24).

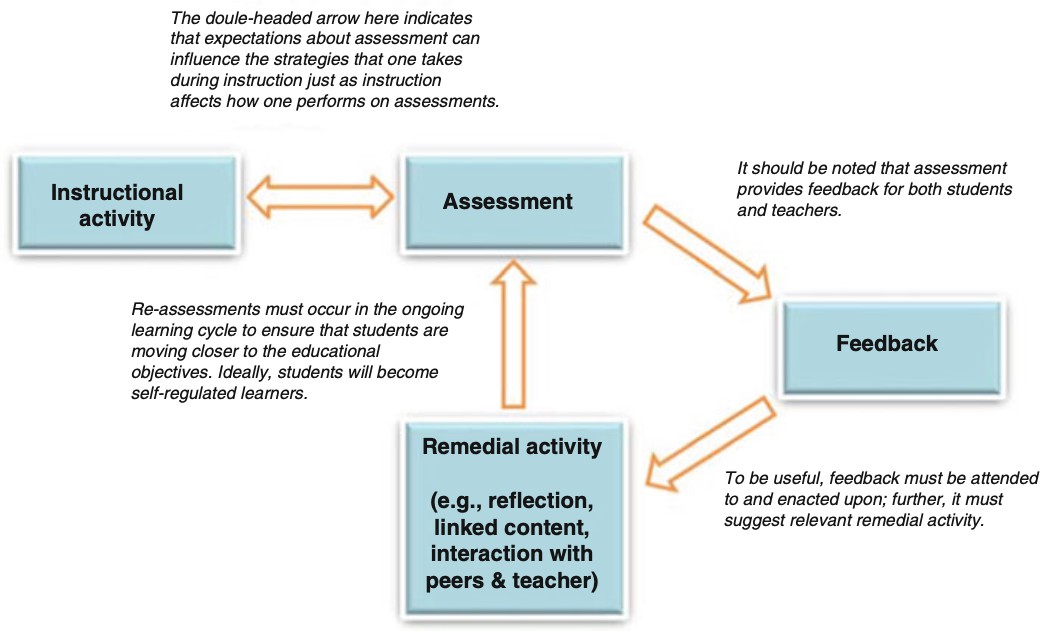


Figure 2.4: A formative assessment process cycle. Reproduced from Trumpower et al. [[1](#_bookmark134)].

### Criteria for Effective Formative Assessment

As summarized in Table [2.1](#_bookmark26), scholars have made various efforts to examine whether a formative assessment is effective from different angles.

One of the most prevailing frames is whether assessment is able to provide specific and useful feedback that supports the improvement of students’ learning. Besides the criterion of feedback, the ability of assessing students’ understanding of higher-order knowledge is another angle of evaluating the effectiveness of formative assessment. Shepard holds the view that valid formative assessment should address conceptual understandings of higher-order knowledge [[31](#_bookmark164)], which aligns with NRC’s recommendation that effective assessment should

be able to assess students’ conceptual and structural knowledge [[27](#_bookmark160), [32](#_bookmark165)]. In addition, the ability of identifying students’ specific strengths and weaknesses and the generalizability of application in learning assessment are considered as important criteria for effective formative assessment [[1](#_bookmark134), [33](#_bookmark166)].

##### Study Focus Criteria

Black et al. [[33](#_bookmark166)] Assessment of learning Easy to carry out, generalizability

Boston [[34](#_bookmark167)] Concept of formative assess-

ment

Being specific, feedback

National Research Council [[27](#_bookmark160), [32](#_bookmark165)]

Assessment of learning Assess students’ conceptual and

structural knowledge

Gibbs & Simp- son [[35](#_bookmark168)]

Haug & Øde- gaard [[36](#_bookmark169)]

Conditions under which as- sessment facilitate learning

Formative assessment and teachers’ sensitivity to stu- dent responses

Assessment task, characteristics of feedback, students’ reactions to feedback

High level of pedagogical knowl- edge, students’ responses to feed- back

Shepard [[31](#_bookmark164)] Validity of formative assess-

ment

Higher-order and transferable un- derstanding of knowledge

Trumpower, Filiz, & Sarwar [[1](#_bookmark134)]

Assessment for learning us- ing knowledge map

Higher-order knowledge, students’ strengths and weaknesses, feedback that improves learning, and user

friendly Table 2.1: An overview of research on criteria for effective formative assessment.

In Section [2.2.3](#_bookmark27), I discuss reasons why knowledge maps and *computer-aided* mapping techniques are used in formative assessment and how they meet the criteria above as e*ffective and reliable* tools in formative assessment.

### Reasons of Using Knowledge Maps in Formative Assessment

##### Benefits of Using Knowledge Maps

Studies show that there are various noteworthy improvements that benefit students when constructing knowledge maps in learning [[1](#_bookmark134), [2](#_bookmark135), [3](#_bookmark136), [4](#_bookmark137), [5](#_bookmark138), [28](#_bookmark161), [37](#_bookmark170), [38](#_bookmark171), [39](#_bookmark172), [40](#_bookmark173)]. The construction of knowledge maps is an active learning process that promotes students’ meaningful reflection and encourages their metacognition.

First, as discussed in Section [2.1](#_bookmark15), constructing a knowledge map is a prevalent approach of externalizing learners’ internal mental models, which allows learners to depict their knowl- edge in external models. By doing so, it not only allows learners to visualize multiple cause– effect patterns and their correlations between different concepts and ideas, it also provides learners with a unique opportunity to recognize their decision-making patterns and how they choose the pathways in problem-solving [[5](#_bookmark138)].

Second, compared to text-based assessments (e.g., argumentation and essay), the visual- ization function of knowledge maps offers an easier method for learner to navigate a content space instead of using keyword search, which requires higher-level skills [[41](#_bookmark174)]. In addition, as a learning scaffold, knowledge maps could deepen learner’s understanding of content by break- ing a topic into manageable chunks [[42](#_bookmark175)]. This allows learners to concentrate on particular segments of the topic, which reduces learners’ cognitive load during the learning process [[5](#_bookmark138)]. Third, when students build knowledge maps individually, it facilitates students’ con- ceptual understanding of knowledge, and the map-modified process could be used to gauge changes in students’ understanding of knowledge [[5](#_bookmark138)]. When students collaboratively con- struct a knowledge map, they are able to gain a deeper understanding of concept relation- ships in a certain knowledge domain. Novak emphasizes that students are conceptualizing and constructing metal models to represent the information they learn when constructing knowledge maps, whether individually or collectively [[2](#_bookmark135)]. The process of identifying, defin- ing, and refining concepts in the model of knowledge maps promotes students’ higher-level

thinking skills needed to understand higher-order knowledge [[43](#_bookmark176)].

Last but not least, the knowledge maps created by students could be used by the in- structor to record and trace students’ understanding of concepts, because knowledge map- ping techniques are regarded as representation of students’ knowledge structures and thus provide a possible way to mediate student’s conceptual knowledge construction [[40](#_bookmark173)]. Well- connected knowledge structures in the form of a knowledge map can offer instructors feedback regarding how well students externalize knowledge in a specific knowledge domain as well as how well they understand the concepts. In addition, the tool of knowledge maps could be used to detect, identify, and address errors, misconceptions, and knowledge gaps presented in students’ knowledge structure [[28](#_bookmark161)].

##### Reliability of Using Knowledge Maps in Formative Assessment

Typically, research on the use of knowledge maps in formative assessment examines the effectiveness and reliability in assessing content knowledge and the application of knowledge maps as instructional tools that provide feedback to enhance students’ understanding in certain knowledge domains. In this session, I discuss evidence of using general knowledge maps and computer-aided mapping techniques in particular in formative assessment.

Based on the discussion in Section [2.2.2](#_bookmark25), the four main aspects of criteria for effec- tive formative assessment are: providing learners with useful feedback, assessing learners’ understanding of higher-order knowledge, identifying learners’ specific strengths and weak- nesses, and the generalizability of application. As shown in Table [2.2](#_bookmark28), studies have provided suﬀicient evidence on how knowledge maps meet the above criteria for effective formative assessment.

* ***Providing useful feedback***: Ifenthaler’s study shows that students present greater development when they are prompted to reflect on the similarities and differences between their own structural knowledge maps and those from experts [[44](#_bookmark177)]. Similarly, Herl et al. state that knowledge maps are able to facilitate students’ learning when feedback is provided on the construction of students’ maps [[9](#_bookmark142)]. Also, students report

**Criteria Evidence Studies**

Students learn effectively when similarities and differences

Providing useful feedback

Assessing understanding of higher-order knowledge

Identifying strengths and weaknesses

Generalizability of

between experts’ knowledge structure and their own knowledge structure are provided.

Computer-based knowledge mapping system with feedback facilitate students’ understanding of content.

Students learn more effectively with additional instructions in map construction.

Various knowledge map scoring methods have been widely applied and proved to be valid.

Evaluation of students’ knowledge maps at different points is associated with educational outcomes.

The process of constructing knowledge maps helps students to identify misconceptions in understanding concepts.

Identify the gap between the conceptual models of students and experts.

The flexibility of knowledge maps. It can be used as an

[[44](#_bookmark177)]

[[9](#_bookmark142)]

[[45](#_bookmark178)]

[[46](#_bookmark179)]

[[47](#_bookmark180)]

[[48](#_bookmark181)]

[[3](#_bookmark136)]

application

in-class activity, individual or group assignment. [[47](#_bookmark180)] Report satisfaction with knowledge map-based formative assessment systems.

Table 2.2: Knowledge maps meet the criteria for effective formative assessment. Adapted from Trumpower et al. [[1](#_bookmark134)].

that they understand and appreciate feedback regarding map construction and learn more effectively with additional instructions in their map construction process.

* ***Assessing higher-order knowledge***: Strautmane presents a literature study on effective usage of knowledge map-based tasks for assessing knowledge purpose, which shows that map-based procedures are a valid measure of higher-order knowledge [[46](#_bookmark179)]. As well, Buldu and Buldu demonstrates that the evaluation of students’ knowledge maps can be used to document and depict students’ learning improvement at different points [[47](#_bookmark180)].
* ***Identifying strengths and weaknesses***: Both studies from Hasemann & Mansfiel and Giabbanelli & Tawfik show that knowledge maps are capable of identifying stu- dents’ learning strengths and weaknesses by detecting correct, incorrect/missing links in the maps respectively [[3](#_bookmark136), [48](#_bookmark181)].
* ***Generalizability of application***: The survey conducted by Buldu and Buldu proves that knowledge maps are flexible to be incorporated in different aspects of assessment activities as a useful tool [[47](#_bookmark180)].

## How to Assess Maps?

As discussed in Sections [2.1](#_bookmark15) and [2.2](#_bookmark22), a knowledge map is a graphical mental model and tool that could be used to externalize and represent students’ knowledge structure in certain domains, and the mapping technique is widely applied in learning assessment. The scholarship reviewed in Section [2.2](#_bookmark22) has presented suﬀicient evidence that the knowledge map and its mapping techniques are effective and reliable in formative assessment.

In Section [2.3](#_bookmark29), I present an overview of how to assess maps in general, and I focus primarily on those that support *automated evaluation of computer-based knowledge maps*. Specifically, Section [2.3.1](#_bookmark30) introduces three properties of knowledge maps as an assessment tool: task format, response format and scoring system. Sections [2.3.2](#_bookmark31) and [2.3.3](#_bookmark33) focus respec- tively on two most prevalent scoring systems of evaluating knowledge maps: reference-free assessment and reference-based assessment. Section [2.3.4](#_bookmark36) reviews a real-world application of hybrid systems (i.e., combining both reference-free and reference-based assessment) in knowledge maps evaluation.

### Properties of Knowledge Maps as an Assessment Tool

As Ruiz-Primo and Shavelson summarize, knowledge maps used as an assessment tool are considered to include: (a) a task that elicits student’s knowledge structure, (b) student’s response format, and (c) a scoring system used to evaluate student’s knowledge maps [[49](#_bookmark182)]. These three different properties of knowledge maps (i.e., task formats, response formats, and scoring systems) may lead to different conclusions and judgement regarding students’ knowledge structures [[4](#_bookmark137), [39](#_bookmark172), [46](#_bookmark179), [49](#_bookmark182)].

##### Task Format

Task format refers to different knowledge map jobs that specify what and how students are expected to complete. Map-based tasks mainly consist of three key components: task demands, task constraints, and the structure of task content [[46](#_bookmark179), [49](#_bookmark182)]. Task demands refer to the expectations placed upon the students, such as filling the missing parts in the map, constructing a map, etc.; task constraint defines the protocols and instructions that students need to follow to complete the task; and the structure of task content means a map structure, such as hierarchical, chain-like maps, etc.

Different combinations of these three components would construct different tasks that serve different purposes. For example, high-directed tasks with strict constraints and a spec- ified structure (e.g., selecting potential concepts from a given list to construct a hierarchical concept map) give students less space to freely express their own knowledge structures, which may restrain students in interpreting their own knowledge accurately. On the contrary, low- directed tasks with free instructions (e.g., constructing concept maps with students’ own glossary) allow students to externalize their knowledge more precisely [[46](#_bookmark179), [49](#_bookmark182), [50](#_bookmark183)].

##### Response Format

Response format may vary from knowledge maps created by pen and paper to those generated via computer-based techniques. According to Ruiz-Primo and Shavelson, three components are identified in response format: response mode (e.g., responses generated with pen and paper, oral expression, or computer-based methods), characteristics of response format (e.g., responses along with a list of terms provided by the mapper), and the mapper (the one who creates the map) [[49](#_bookmark182)]

Compared to maps created with pen and paper, computer-based maps are easier to mod- ify and revise. Thus, they are conducive for developing automated generation and assessment mechanism for knowledge maps. One drawback of using computer-based techniques is that this requires essential training for instructors and students on how to use certain computer- based software to generate knowledge maps [[4](#_bookmark137)].

##### Scoring Systems

A scoring system refers to a systematic method that evaluates students’ knowledge maps consistently and accurately [[4](#_bookmark137), [49](#_bookmark182)]. Due to the large variety of map-based tasks and responses to those tasks, it seems that there is no one single best scoring system that reflects stu- dents’ knowledge structures precisely. Rather, myriad scoring systems have been applied in research. These systems can be classified into three most general scoring systems: reference- free scoring systems, reference-based scoring systems, and a combination of reference-free and reference-based scoring systems (i.e., hybrid systems).

***Reference-free scoring systems*** employ different scoring criteria to measure the quantity and quality of different components of knowledge maps, such as the number of concepts, proposition quality, the number of propositions, etc. Section [2.3.2](#_bookmark31) provides detailed descriptions of this type of scoring system.

***Reference-based scoring systems*** involve an additional criterion map (i.e., expert map), compare the student’s map with the criterion map, and score the differences and similarities between them. Section [2.3.3](#_bookmark33) summarizes this approach.

***Hybrid systems*** combine the approaches of assessing map components and comparing the map with a reference/criterion map. Section [2.3.4](#_bookmark36) provides a detailed illustration of using hybrid systems for real-world problems.

### Reference-free Scoring Systems

Reference-free scoring systems often treat and assess a knowledge map as a graph, where many components and structural measures are computed based on a graph [[7](#_bookmark140), [4](#_bookmark137), [10](#_bookmark143), [5](#_bookmark138), [11](#_bookmark144)]. A graph is considered as: *G* = (*V, E*), where *V* is the set of vertices, and

*E* ⊆ {(*x, y*)|*x, y* ∈ *V, x* /= *y*}, a set of edges that links the vertices.

In the knowledge map, vertices represent the nodes or points of the map, while the edges

represent the links from a concept onto another. The analysis of a graph that is used in

assessing knowledge maps could be achieved via two aspects: vertex level and graph level. Some terminologies of graph theory are explained below:

##### Vertex level: centrality measures for a single vertex of a graph

* ***Degree***: total number of edges that are indent to the vertex [[51](#_bookmark184)], which is defined as:

*deg*(*u*) = |(*u, v*) ∈ *E*(*G*)*,* for all v ∈ *V* (*G*)| (2.1)

* ***Closeness***: the closeness of a vertex describes how close a vertex is to every other vertex [[51](#_bookmark184)]. If the sum of the distances is larger, then the closeness is smaller. The closeness of a vertex is defined as:

*C*(*u*) = Σ

*u*̸=*w*

1

*d*(*w, u*)

(2.2)

where *duw* is the length of the shortest path between vertices *u* and *w* of a graph *G*.

* ***Betweenness***: the betweenness of a vertex represents the importance of a vertex within a graph. The betweenness of a specific vertex is equal to the number of shortest paths that pass through that vertex for all pairs of vertices in the graph [[51](#_bookmark184)]. The betweenness of a vertex is defined as:

*B*(*v*) =

Σ

*u*̸=*w*̸=*v*

*σuw*(*v*) *σuw*

(2.3)

where *σuw* is the total number of the shortest path between vertices *u* and *w* of a graph *G* and *σuw*(*v*) is the total number of the shortest paths between vertices *u* and *w* that pass through vertex *v*.

##### Graph level: measures for the whole graph

* ***Order***: total number of vertices |*V* (*G*)|.
* ***Size***: total number of edges |*E*(*G*)|.
* ***Density***: number of edges existing as a ratio of maximum edges 2|*E*(*G*)| .

|*V* (*G*)|∗(|*V* (*G*)|−1)

* ***Number of cycles***: A cycle in a graph is denoted as: a non-empty trail (*e*1*, e*2*, . . . , en*) with a vertex sequence (*v*1*, v*2*, . . . , vn, v*1), where *e*1*, e*2*, . . . , en* ∈ *E*(*G*) and *v*1*, v*2*, . . . , vn* ∈ *V* (*G*).
* ***Diameter***: the furthest length between two vertices of the graph [[51](#_bookmark184), [4](#_bookmark137)]. It is denoted as:

*diameter*(*G*) = *max*(*duw*) : *u, w* ∈ *V* (*G*)*, u* /= *w* (2.4)

where *duw* is the distance of the shortest path between vertex *u* and *w* of a graph *G*.

* ***Compactness***: Average of the distances between all vertices. Higher values indicate

a lower compactness [[4](#_bookmark137), [51](#_bookmark184)]. It is denoted as:

*compactness*(*G*) = 1 *n*(*n* − 1) *d*

Σ

2 *uw*

: *u, w* ∈ *V* (*G*)*, u* /= *w* (2.5)

where *duw* is the distance of the shortest path between vertex *u* and *w* of a graph *G*, and *n* is the total number of vertices |*V* (*G*)|.

* ***Diameter of the spanning tree***: A spanning tree is a portion of the graph where

all nodes are visited once and the visiting path is acyclic. The diameter is the shortest path between the two most distant nodes [[7](#_bookmark140), [51](#_bookmark184)]. It is denoted as:

*γ*(*G*) = *maxi,jdtree*(*i, j*) (2.6)

where *γ*(*G*) refers to the diameter of the spanning tree and *i* and *j* represent two most distant nodes in the spanning tree.

### Reference-based Scoring Systems

Another widely used scoring system is reference-based, which compares a map with a reference/criterion map. The reference map could be considered as the end goal: the student should eventually be able to replicate key features of that map to be deemed correct [[4](#_bookmark137)].

One general way to use the reference map is called *structural matching*, which compares the structural scores by calculating the metrics derived from the graph of both the student and reference map. The similarity value between the student map *S* and the reference map *R* can be expressed as:

|*mS* − *mR*|

*s* = 1 −

*max*(*mS, mR*)

(2.7)

where *ms* and *mr* represent the structural measurement of the student’s and the reference map respectively. If *s* = 0, it means that both maps are completely different; if *s* = 1, it means that the maps are identical [[4](#_bookmark137)]. The user wishes to compute a scoring must supply the structural measure *m*.

Although the assessment method from graph theory produces summative scores based on the map components (i.e., organization of knowledge), none of the structural measures account for the semantic information (i.e., semantic meaning of knowledge), which is a central part in the underlying knowledge representation [[4](#_bookmark137), [52](#_bookmark185)].

Therefore, Ifenthaler argues that, besides the measures that account for the structural metrics of knowledge representation, it is important to incorporate *semantic matching*, such as node content, labels of nodes and links, into assessment. To address the demand of semantic matching and analysis, Ifenthaler proposes applying Tversky Similarity measure into assessment [[52](#_bookmark185)]. The Tversky Similarity measurement [[53](#_bookmark186)] is expressed as :

|*Pa* ∩ *Pb*|

*s* =

|*Pa* ∩ *Pb*| + *α*|*Pa* − *Pb*| + *β*|*Pb* − *Pa*|

(2.8)

where *P* means the property sets of maps (e.g., set of concept, set of propositions), *α* and *β* are weights of different quantities which separate maps *a* and *b*, and they are usually equal to 0.5. The relationship between these two property sets is illustrated in Figure [2.5](#_bookmark35).

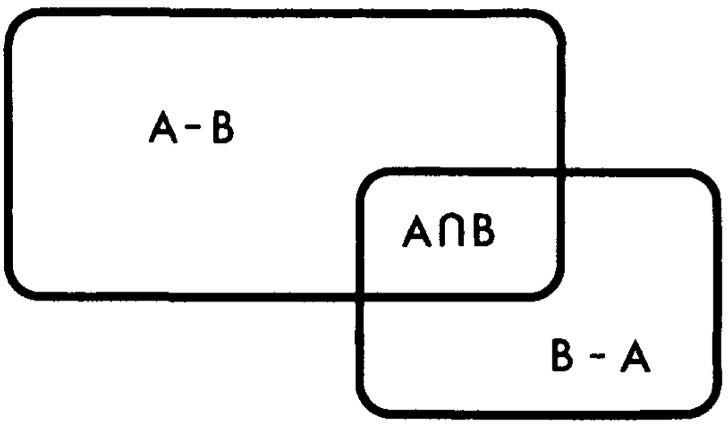


Figure 2.5: A graphical illustration of the relation between two properties sets. Reproduced from Tversky [[53](#_bookmark186)].

### Application of Hybrid Systems

An application of using hybrid systems is called SMD approach (Surface, Matching, Deep Structure). SMD is designed based on the theory of mental model and graph the- ory with the purpose of measuring relational, structural, and semantic levels of graphical representations and concept maps [[7](#_bookmark140)]. It assesses knowledge maps through components of graph (reference-free), structural (reference-free), and semantic matching (reference-based). A specific example provided in this subsection, starting with the transformation of an input model (Figure [2.6](#_bookmark37)).

##### Surface Structure

*Surface structure* represents the relational structure of each model *M* (also a graph), and the simplest way to calculate this metric is counting the total number of propositions *P* (i.e., construct of node-edge-node) in *M* [[7](#_bookmark140)].

*n*

Σ

*θ* = *Pi* : *Pi* ∈ *M* (2.9)

*i*=0

As the example shown in Figure [2.7](#_bookmark38), model *M* only contains two propositions (i.e., cells- consist of-animal cells and cells-consist of-plant cells), while there are three more propositions (i.e., animal cells-contain-nucleus, animal cells-contain-cytoplasma, and plant cells-contain- cytoplasma) included in the reference model. Therefore, the values of *Surface Structure* in model *M* and *Mr* are *θ*(*M* ) = 2 and *θ*(*Mr*) = 6 respectively.

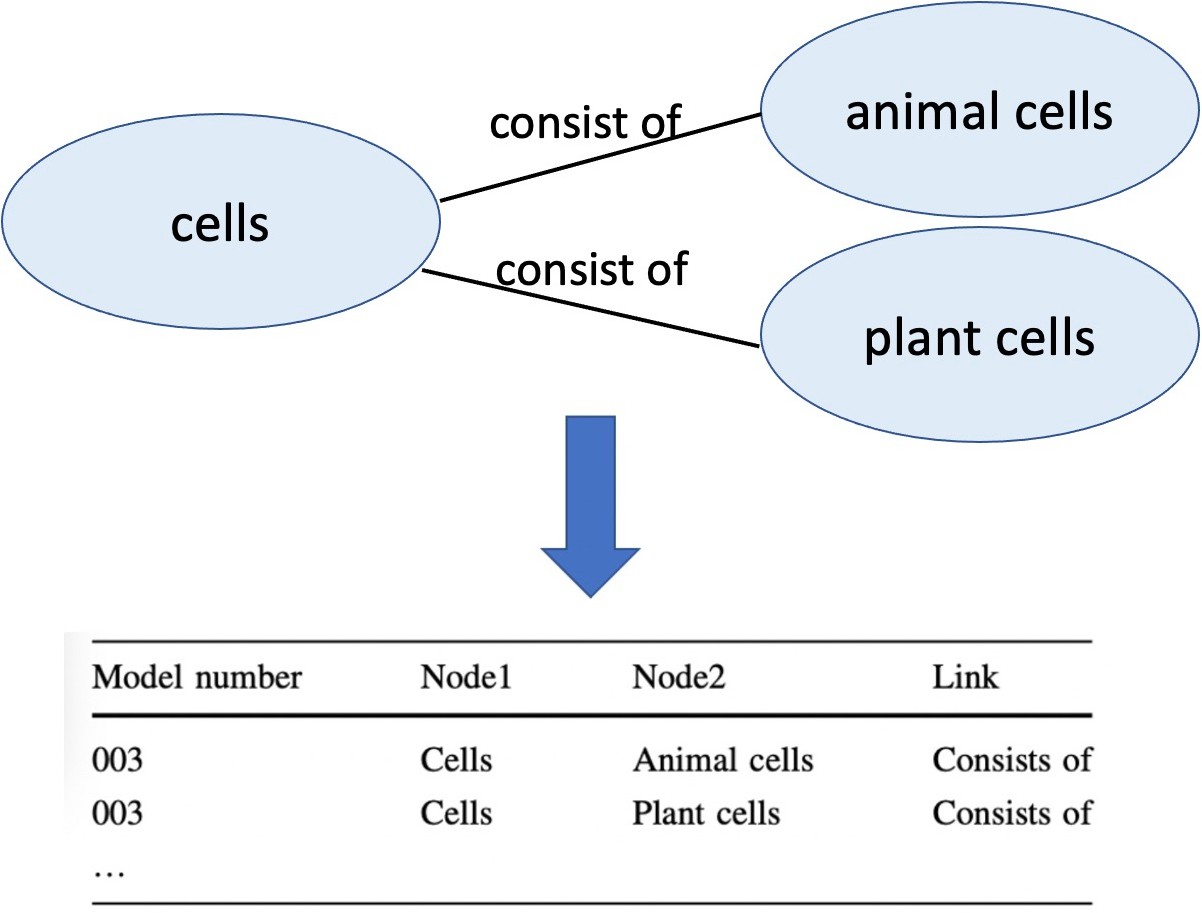


Figure 2.6: Transformation of raw data from a model M to a standardized format. Repro- duced from Ifenthaler [[7](#_bookmark140)].

##### Matching Structure

*Matching structure* represents the structural property of a model *M* and it indicates the complexity of the model *M* [[7](#_bookmark140)]. One metric of matching structure can be computed by calculating the diameter of a spanning tree via Function [2.6](#_bookmark32). As the spanning trees presented in Figure [2.7](#_bookmark38), the *Matching Structure* of model *M* and reference model *Mr* is 2 and 3 respectively. Besides the diameter of a spanning tree, metrics discussed in Section [2.3.2](#_bookmark31), such as graph compactness, node’s closeness, and betweenness, can be also used to describe and analyze the structure of a model M [[7](#_bookmark140)].

##### Deep Structure

*Deep Structure* reveals the semantic information of a model *M* . In contrast to the reference-free evaluation of a graph, model analysis on the *Deep Structure* is realized through a similarity calculation between a model *M* and a reference model *Mr* [[7](#_bookmark140)]. The metric of deep structure is calculated via the Tversky’s similarity function (Function [2.8](#_bookmark34)).

In the case of Figure [2.7](#_bookmark38), if using the set of propositions as the calculating metric, we could get:

* |*Pa* ∩ *Pb*| = 2, which means two common propositions exist in both models.
* |*Pa* − *Pb*| = 0 which means all the propositions that exist in model *M* also exist in the

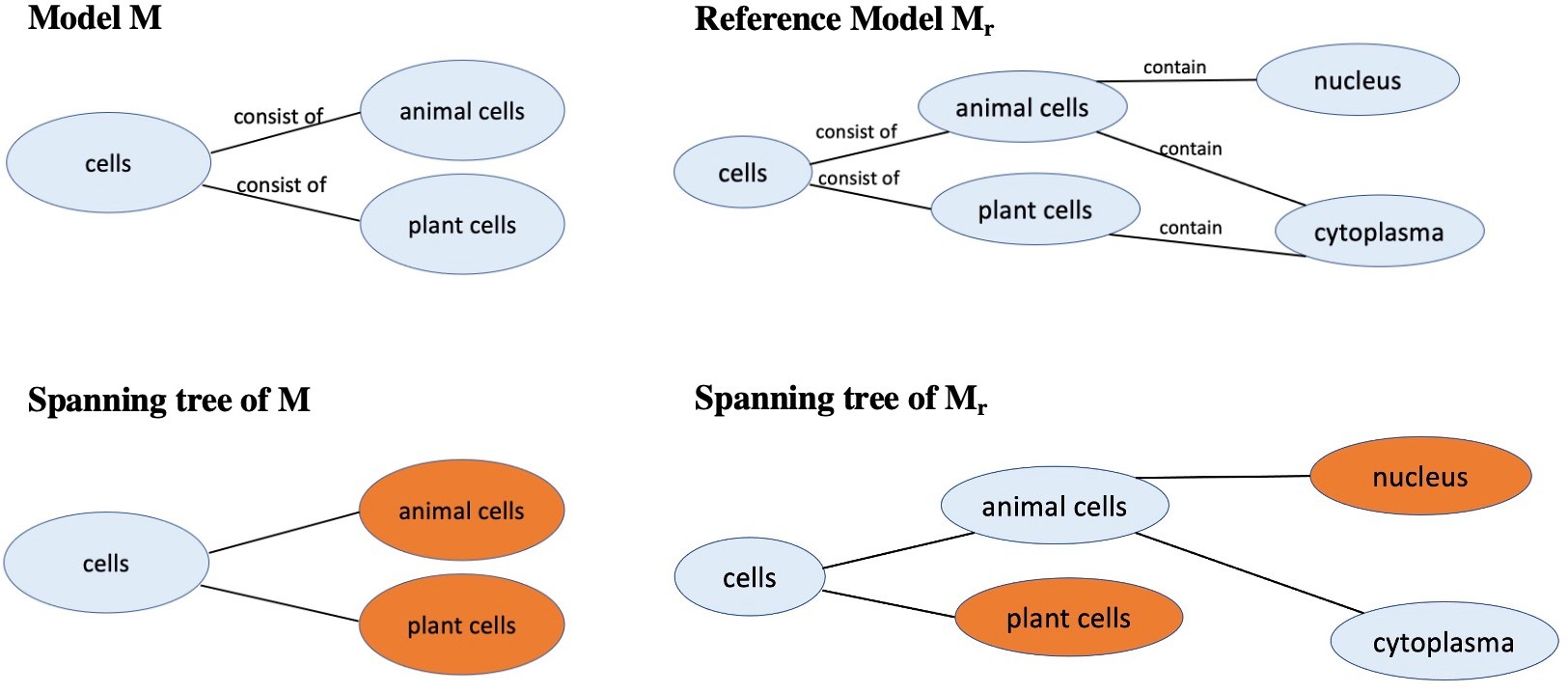


Figure 2.7: Model *M* and reference model *Mr*, and their spanning trees with two most distant nodes marked. Adapted from Ifenthaler [[7](#_bookmark140)].

reference model *Mr*.

* |*Pb* − *Pa*| = 3 which means three propositions exist in the reference model *Mr* but not in model *M* .

When setting *α* and *β* as 0.5, the value of *Deep Structure* is 0.57, which means that the semantic similarity between model *M* and reference model *Mr* is 57%.

## Open Problems in Assessing Maps: Text Mapping, Feedback, and Linguistic Variability

In Section [2.3](#_bookmark29), various assessment theories and methods of knowledge maps have been presented in detail. However, there are still several open problems that exist in the evaluation of knowledge maps, such as converting texts into maps, automatically providing feedback, and addressing the linguistic variability in terminologies that cause alignment issues. There- fore, this section reviews the open issues mentioned above and potential solutions to solve these problems.

### Converting Essays to Causal Structures: Text Mapping

As NRC recommends, an effective assessment should be able to assess students’ concep- tual and structural knowledge [[27](#_bookmark160), [32](#_bookmark165)]. Among existing ways to measure students’ conceptual and structural knowledge, using natural language, particularly students’ writing, is one of most accurate approaches to elicit and represent students’ knowledge structure [[54](#_bookmark187), [55](#_bookmark188)].

However, grading essays is very time consuming for instructors. As Kim states, although automatic writing evaluation tools exist, they are typically used for summative assessment instead of formative assessment [[8](#_bookmark141)]. Thus, it raises the question: can written texts, which derive from an internal mental representation of knowledge model based on students’ inter- pretation of a text, be converted into knowledge maps such that we can apply *automated assessment and feedback system* to support students’ learning?

Many techniques have been proposed to gain insights from text and generate knowledge maps from the insights. A software utility called ALA-Reader was designed to translate students’ written text into raw proximity data, which then could be analyzed by knowledge

network [[56](#_bookmark189)]. The ALA-Reader is able to capture the sequence of important key terms from writing texts as links and store the links in an *n* × *n* matrix, which could be applied in diverse domains and across languages [[8](#_bookmark141), [55](#_bookmark188)]. Pathfinder KNOT is another software that is able to convert raw proximity data generated from ALA-Reader into visualized PFNet rep- resentations (a knowledge-map like representation) [[56](#_bookmark189), [57](#_bookmark190)]. With the support of Pathfinder

KNOT, the most direct path or strongest links between nodes could be identified by simpli- fying complexity and eliminating visual noises of the original network [[55](#_bookmark188), [58](#_bookmark191)].

Graphical Interface of Knowledge Structure (GIKS) is a text analytical tool that in- tegrates key functions of both ALA-Reader and Pathfinder KNOT. The GIKS is able to capture and analyze implicit semantic knowledge structures derived from students’ writing and convert the writing into visual knowledge structure network graphs, in which the most salient linkages among key concepts are presented [[8](#_bookmark141), [55](#_bookmark188), [59](#_bookmark192)]. Besides the ability of con- verting written texts into knowledge maps representation automatically, the GIKS system is also able to compare students’ knowledge structures with the reference knowledge structure to provide information of students’ knowledge strengths and weaknesses [[8](#_bookmark141), [55](#_bookmark188)]. A specific example is shown in Figure [2.8](#_bookmark41). In the example, the left one is a reference KS network derived from a lesson text, and the right one is derived from a student’s writing. The stu- dent’s knowledge strengths and weaknesses are shown in green, yellow, and red links, which represent the similarities and differences compared to the reference map.

Other tools include Concept Map Miner (CMM) and its variation that integrates CMM within Glossor (i.e., a tool for automated feedback that prompts reflection on writing). They could not only create a visualization environment that automatically generates concept maps from the text using text mining algorithms, but also are able to provide automated feedback on the writing [[60](#_bookmark193), [61](#_bookmark194)].

While the above approaches (i.e., ALA-Reader, Pathfinder KNOT, GIKS, CMM) are able to extract information and generate knowledge maps from text, those maps still have some important limitations. For example, the main drawback of GIKS is its limited application scope in that only fairly technical texts or specific vocabulary could be analyzed within GIKS. However, less technical text or synonymous terms may not be recognized, thus leading to increased errors [[8](#_bookmark141)]. The maps representation from CMM are often presented in a “still” state of what is in the text itself, which does not highlight dynamics events [[61](#_bookmark194)].

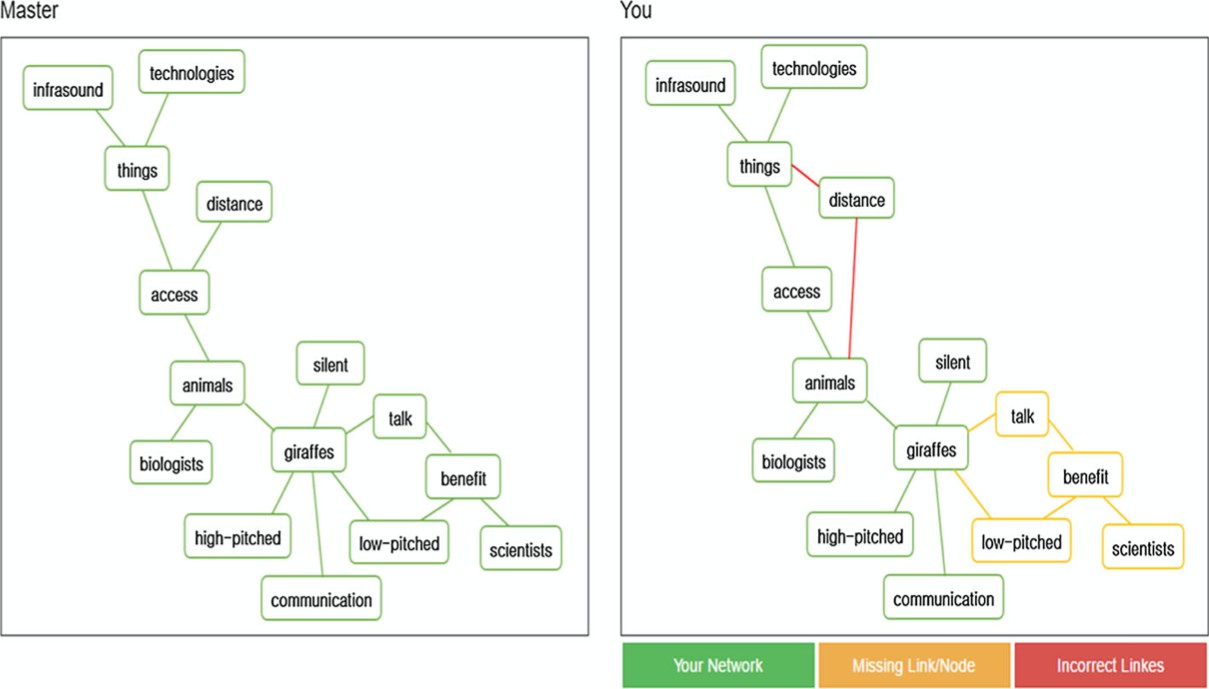


Figure 2.8: Reference and student map generated by GIKS. Reproduced from Kim [[8](#_bookmark141)].

### Automated Feedback

Feedback is essential to the learning process, and it is also a critical part of formative learning [[3](#_bookmark136), [12](#_bookmark145), [28](#_bookmark161), [60](#_bookmark193)]. For effective assessment, instructors are not only expected to point out students’ misconceptions, but also hope to provide feedback and solutions to help students to improve. The cycle of evaluation-feedback-modification has been proved to significantly improve students’ learning and understanding of knowledge in certain domains [[6](#_bookmark139)]. However, providing detailed feedback requires additional time and extra work for instructors [[3](#_bookmark136), [6](#_bookmark139)]. Thus, this raises another question: is it possible to integrate automated feedback system into the computer-based knowledge map assessment tools?

In Section [2.4.1](#_bookmark40), two tools that are able to automatically generate feedback have been mentioned: GIKS and CMM within Glossor. However, as shown in Figure [2.8](#_bookmark41), the feedback information that GIKS provides is very limited. The feedback only highlights differences, but does not provide further information of how to bridge the gap between the student map and reference map. For CMM with Glossor, the feedback generated takes into effect only when students start analyzing the maps along with reflecting on guidance questions, such as “do you think the most relevant concepts you covered in the essay are present in the map?” [[60](#_bookmark193)].

In this section, I focus on examining another software called Incremental Thesaurus for Assessing Causal Maps (ITACM), which is designed to automatically generate feedback as well as to assess maps. The current version of ITACM (version 4) offers several unique

features: 1) compare student and expert maps through both interactive visualizations and structural analyses; 2) the repository of terminologies keeps growing by remembering stu- dents’ variations over multiple analysis sessions and analysis across students; 3) the em- bedded structure–behavior–function (SBF) framework allows users to control the level of concept complexity; 4) various approaches to measure the similarity of maps (graph kernel, graph editing distance, graph embedding); 5) the ability to automatically generate feed- back [[3](#_bookmark136), [13](#_bookmark146), [37](#_bookmark170)].

When designing the function of automatically generating feedback, ITACM follows six principles below [[3](#_bookmark136)]:

1. The terminologies used by the students should be preserved in system for future use.
2. Students should hold the responsibility of making changes and modification when they are given enough hints.
3. Students should be exposed to limited but most critical parts that need changes.
4. Minimal steps are expected to narrow the gap of knowledge structure (represented as maps) between the student and the expert.
5. The goal to bridge the gap between the student map and expert map should be achieved in continual changes instead of jumping across the map.
6. The software interface should be user friendly.

An implementation of generating feedback from ITACM is shown in Figure [2.9](#_bookmark44). In a), suggested changes are stored in a table that helps transform a student’s map into the expert’s map. Each change action is linked with modifiable rationals in it. Two parameters (maximum length of loops and disjoint paths1) are used to control the feedback list. In b), feedback on the corresponding part of the map is highlighted with color red and reasons for the change are also available when clicking the edge.

### Linguistic Variability

Another challenge of assessing knowledge maps is due to the linguistic variability that results in different terms used in a student’s map and an expert map even for the same knowledge construct [[13](#_bookmark146)]. An example is showed in the Figure [2.10](#_bookmark45). When assessing the knowledge network (a) with the expert network (b), the meaning of terms in student’s network like “knows company’s value” and “knows company’s belief” should be considered as alignment with the term “good cultural fit” in the expert network.

Therefore, solving the term alignment problem is an essential step to effectively assess students’ knowledge constructs presented in maps. This section revisits the ITACM software discussed in Section [2.4.2](#_bookmark42) and its approach to solving the alignment issue. In addition, this

1A path represents a logical series of steps or “actions” for the problem domain. Given an action, there can be several consequences, each leading to its own path. The number of disjoint paths thus captures the number of alternative consequences that are expected in the student’s map. Increasing the number of alternatives can get to the point where a student would be expected to have thought of plausible, yet unlikely, effects of an action.

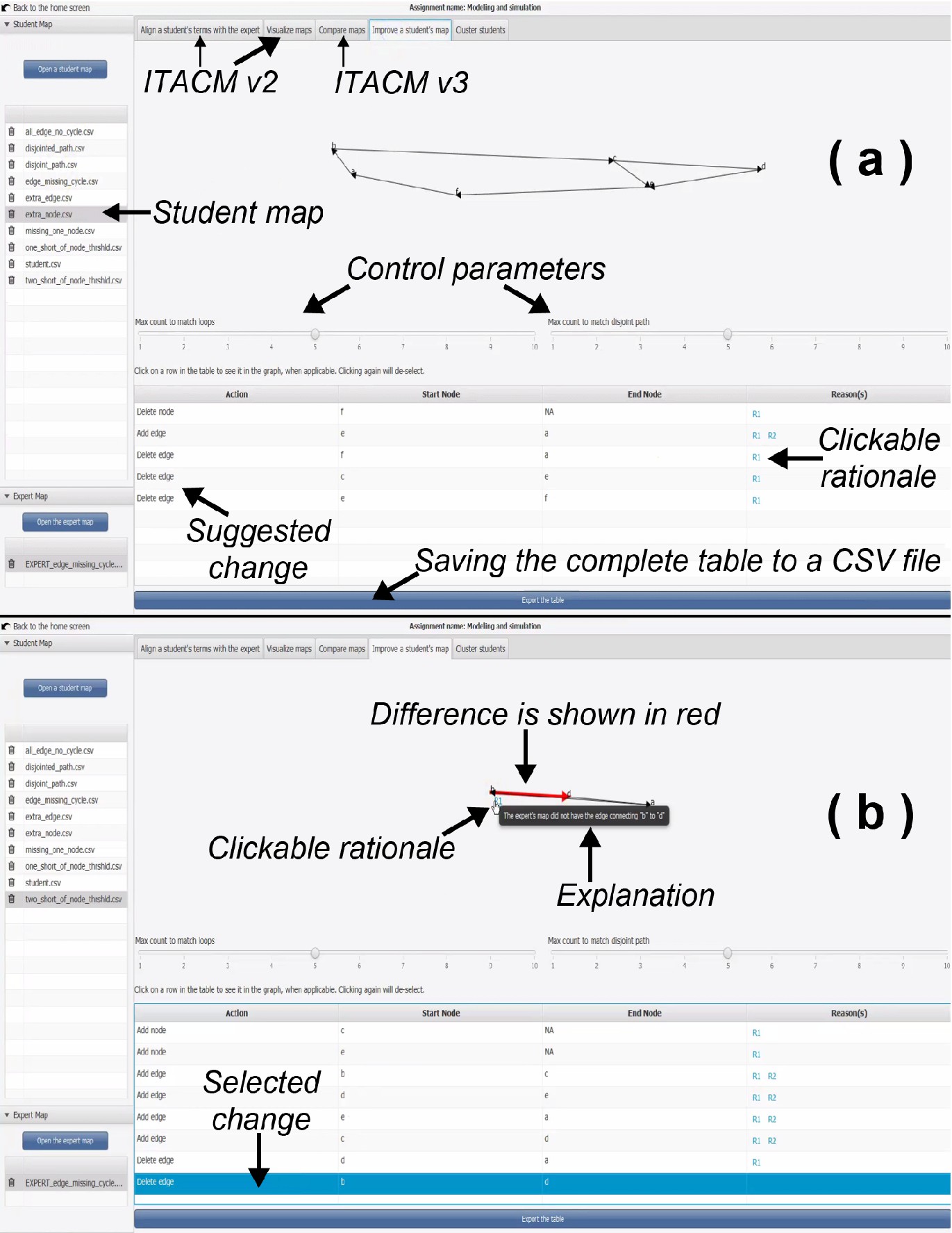


Figure 2.9: Implementation of ITACM in generating feedback. Reproduced from Giabbanelli and Tawfik [[3](#_bookmark136)]. 26

section investigates one familiar applied approach in reducing the semantic gap between different overlapping terminologies in the same domain: ontology matching.

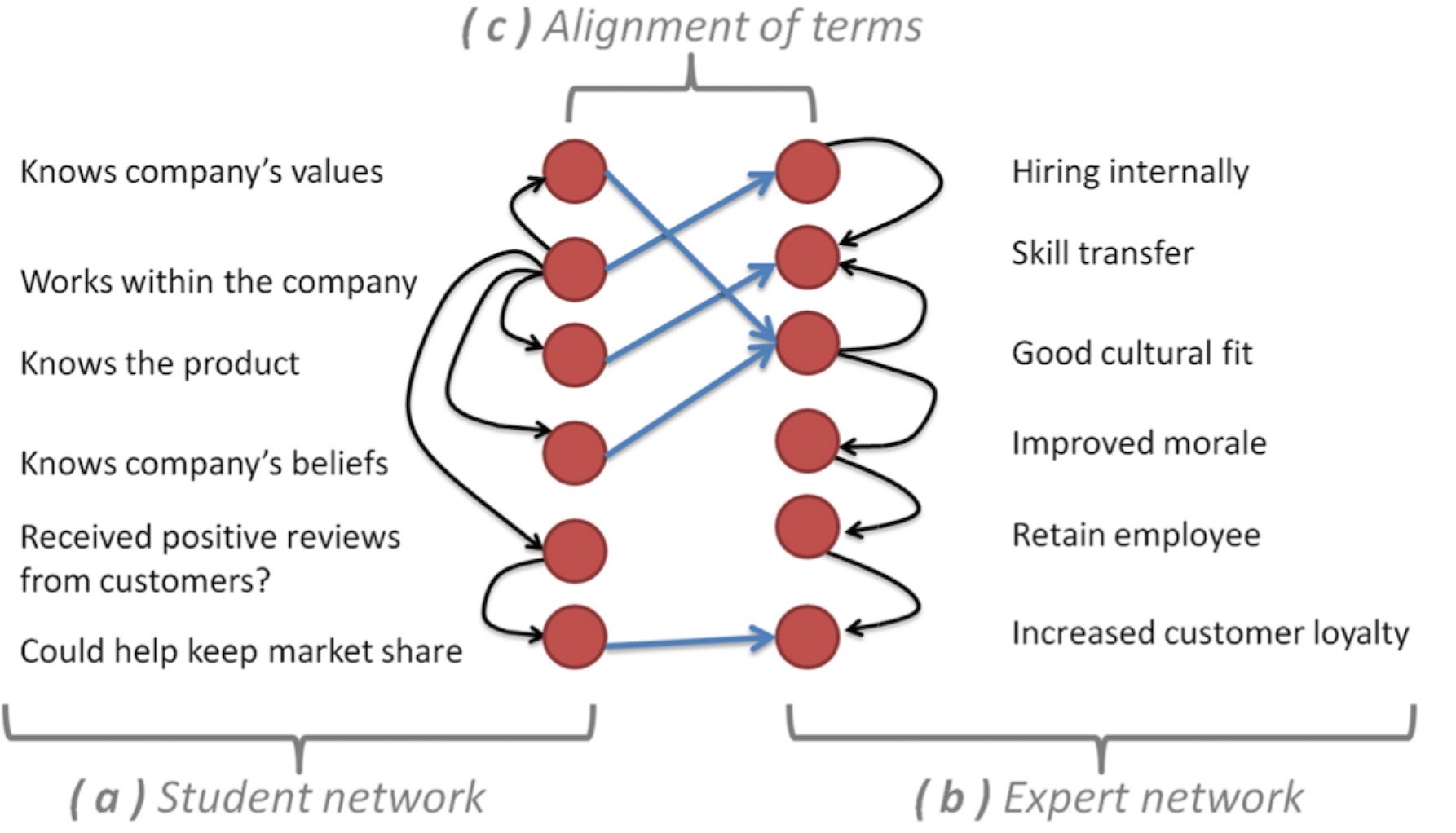


Figure 2.10: Terms in the student’s network (left) may not be exactly the same as the expert’s network (right), but they may mean the same knowledge representation. Figure is reproduced from Gupta, Giabbanelli and Tawfik [[13](#_bookmark146)].

##### ITACM Alignment

ITACM relies on an online environment where assignments and terminology alignments are able to shared. The repository of terminologies are managed collaboratively, and the variations of terminologies over multiple analysis sessions and analysis across students are stored in databases and shared with the instructor’s community [[3](#_bookmark136), [13](#_bookmark146), [37](#_bookmark170)].

The alignment is achieved by a recommendation algorithm, which is based on a weighted aggregation method instead of Natural Language Process. An alignment will be recom- mended based on three principles: 1) terms endorsed by other users 2) exact same term,

3) favoring users from the same application context [[13](#_bookmark146)]. The recommendation algorithm is denoted as follow:

*W* (*A*) = *users*(*a, d*) *if d* /= *domain*(*u*) *users*(*a, d*) ∗ *α if d* = *domain*(*u*)

Σ Σ

(2.10)

*a*∈*A d*∈*D*

where *A* means a list of alignments, *D* is a list of domains, *u* is a user, *W* (*A*) is the total weight of all alignments, and *α* is a preferential weight that favors alignments in the same

domain. The weight *w*(*a*) of an alignment where *a* ∈ *A* is the weighted summation of the users having this alignment, which is normalized by the total weight as bellow:

Σ*d*∈*D*

*users*(*a, d*) *if d* /= *domain*(*u*) *users*(*a, d*) ∗ *α if d* = *domain*(*u*)

*W* (*A*)

(2.11)

However, there is one big drawback of this alignment technique: the system would only offer recommendation to a term if it is exactly the same in the previous alignment. Any small alteration of the term would cause a new calculation, thus resulting in a different alignment [[13](#_bookmark146)]. For example, apple and apples would represent different terms in the recom- mendation system, while in reality they represent the same knowledge construct based on contexts. In the following part, ontology matching, a more frequently used and complex ap- proach to reduce the semantic gap between overlapping representation in the same domain, is reviewed.

##### Ontology Matching

An ontology usually specifies a vocabulary that “describes a domain of interest and a specification of the meaning of terms used in the vocabulary” [[62](#_bookmark195)]. Based on the specification, the ontology may consist of different sets of terms, classifications, thesauri, or database schemas. Ontology matching is a widely used technique in solving semantic heterogeneity problems. The mechanism of ontology matching is searching for semantic correspondences related entities between two different ontologies [[62](#_bookmark195), [63](#_bookmark196), [64](#_bookmark197)].

According to Euzenat and Shvaiko [[63](#_bookmark196)], a correspondence is defined as:

Given two ontologies *O*1 and *O*2, a correspondence among entities *e*1 and *e*2 from *O*1 and *O*2

respectively is a five tuples:

*< id, e*1*, e*2*, r, n >*

such that:

* *id* is an identifier for the given correspondence;
* *e*1 *and e*2 are entities, e.g., classes and properties of the first and the second ontology respectively;
* *r* is a relation, e.g., equivalence (=), more general (≥), less general (⊆) between *e*1 and

*e*2;

* *n* is a confidence measure between *e*1 and *e*2 (typically range from 0 to 1).

Ontology matching are often applied in matching techniques and matching systems, and the basic matching technique can be classified in the sequences shown in Figure [2.11](#_bookmark46), which is proposed by Euzenat and Shavkio [[62](#_bookmark195)]. In addition, information that is used in ontology matching can be classified into four categories: lexical, structural, semantic, and external information [[65](#_bookmark198)].

Based on different matching techniques and information categories, sample ontology matching systems are presented in Table [2.3](#_bookmark47), ontology matching systems in general use

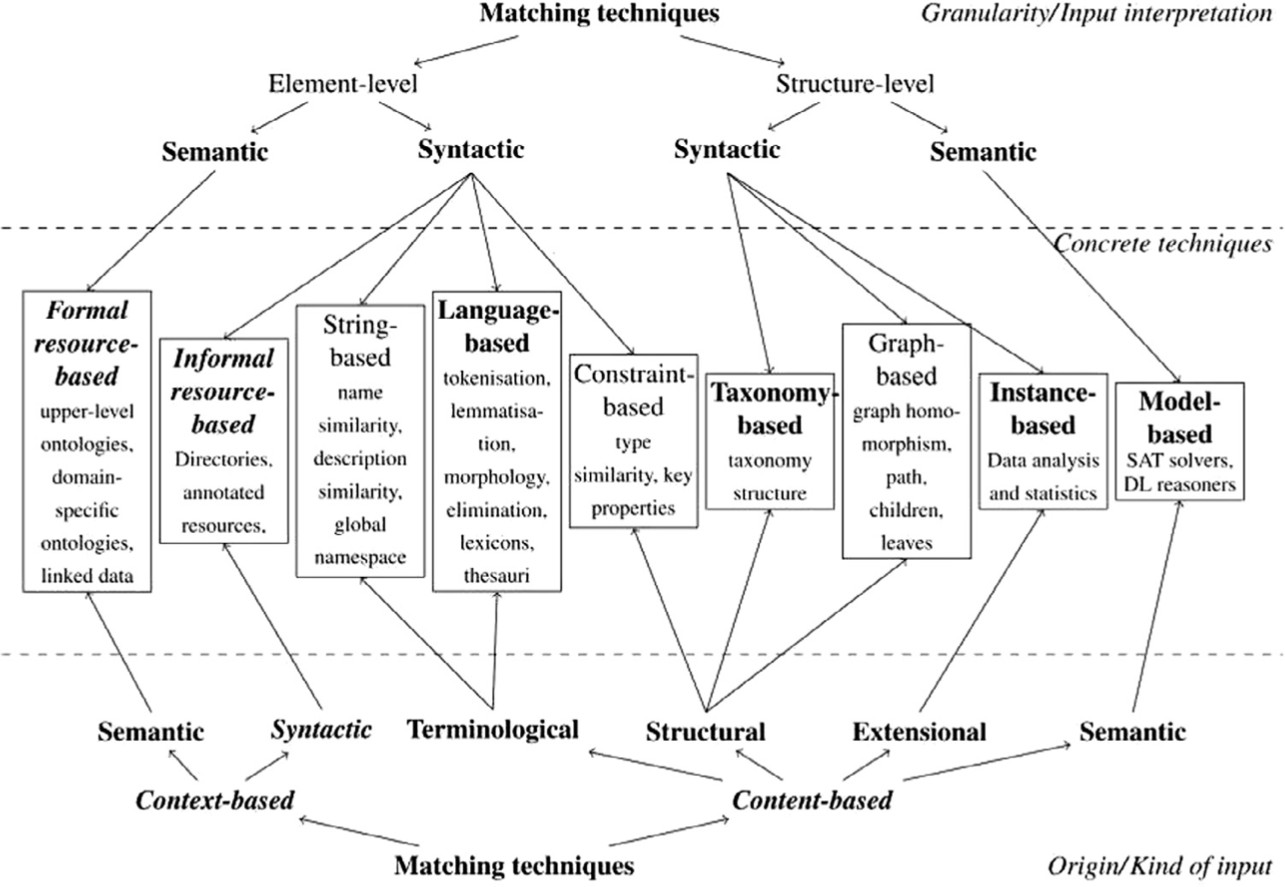


Figure 2.11: Matching techniques classification. Reproduced from Euzenat and Shaviko [[62](#_bookmark195)].

##### Applications Information Categories Matching Techniques

Lily [[66](#_bookmark199), [67](#_bookmark200)] lexical, structural language-, graph-, and model-based ASMOV [[68](#_bookmark201), [69](#_bookmark202)] lexical, structural, external resource-, string- and taxonomy-

based

RiMOM [[70](#_bookmark203), [71](#_bookmark204), [72](#_bookmark205)] lexical, structural, external string-, resource-, and instance-

based

iMatch [[73](#_bookmark206)] structural graph-, model-, and constraint- based

S-Match [[74](#_bookmark207), [75](#_bookmark208)] lexical, semantic , external language- and graph-based

Medto [[76](#_bookmark209)] structural and external resource-, model-, and graph-based Table 2.3: A summary of different ontology matching systems.

more than one type of information in the matching process and incorporate at least two matching techniques in ontology alignment. When the system needs to examine lexical in- formation, language-based or string-based techniques are typically applied, such as Lily, AS- MOV, RiMOM, and iMatch. The frequently used string-based methods include the edition distance-based method, and the prefix/suﬀix similarity method, etc., and language-based techniques mostly rely on Natural Language Processing [[64](#_bookmark197), [65](#_bookmark198)]. In addition, the analysis of

structural information often comes with graph-based matching or model-based techniques, such as Lily, ASMOV, S-Match, and Medto. Graph-based techniques often consider the ontologies as labelled graphs and treat the matching problem as a graph homomorphism problem, while model-based techniques achieve the goal of alignment by exploiting the se- mantic interpretation linked to the input [[64](#_bookmark197)].

## Summary

In summary, this chapter has reviewed scholarship in relation to the three questions that guide my research topic on applying knowledge maps in learning assessment: 1) How to elicit and represent knowledge as a graph/network? 2) Why knowledge maps are useful in assessments for learning? and 3) How to assess maps? It has also closely examined the open problems existing in current literature and potential solutions to address those problems.

**Chapter 3**

# Identifying Informative Features to Evaluate Student Knowledge as Maps

Knowledge maps have been widely used in knowledge elicitation and representation to evaluate and guide students’ learning. To effectively evaluate maps, instructors must select important and relevant map features that capture students’ knowledge constructs. However, there is currently no clear and consistent criteria to select such features, as empirical studies continue to reflect the (implicit) preferences of scholars. This is challenging for instructors, who may thus ignore critical aspects of a map and/or waste their efforts by examining highly correlated features. To address the research gap of selecting informative graph metrics for assessment of knowledge maps, we adopt the machine learning technique of Unsupervised Feature Selection (UFS). Specifically, we extract 12 features used in the prior literature on map assessment (e.g., density, diameter) and use 8 UFS algorithms to rank their importance. By using 202 maps originating from four case studies, we identify features that are generally (un)informative and observe nuances due to context (e.g., learning task, participant profiles). Results suggest that features commonly reported (e.g., number of edges) may not be as informative as less commonly examined aspects (e.g., average degree). Differences exist between maps: the diameter is valuable when learners produce maps from detailed studies, but less informative when maps are directly elicited from the learners’ perspectives. The 8 UFS algorithms show five distinct ways to rank features in maps, hence future work may eliciting the preferences of instructors for grading and map these preferences to an algorithmic approach (i.e., UFS) that produces a ranking.

## Introduction

Based on Ormrod’s definition of learning “as a long-term change in mental representations or associations as a result of experience” [[14](#_bookmark147)], the learning entity *resides implicitly in a person’s mind*, which makes it impossible to assess learning *directly*. In addition, complex problems faced by learners are often open ended, which means that they need to have a detailed understanding of the problem space in the learning process to make and explain their decisions. Thus, knowledge elicitation and representation become essential when it comes to evaluating and guiding either the learning *process* or the learning *outcomes* of learners. Knowledge externalization can be realized through several ways, such as speaking aloud, writing a text, drawing a picture, or constructing knowledge maps [[7](#_bookmark140)]. Various network- based conceptual models, including Novakian concept maps (Fig. [1.1](#_bookmark6)-a), causal loop diagram (Fig. [1.1](#_bookmark6)-b), causal map (Fig. [1.1](#_bookmark6)-c), and rich pictures, are frequently applied in eliciting, representing, and assessing knowledge learning [[1](#_bookmark134), [16](#_bookmark149), [17](#_bookmark150)]. The concept of knowledge mapping is widely used in the assessment of students’ learning in the field of education [[1](#_bookmark134), [8](#_bookmark141), [4](#_bookmark137)]. In this chapter, we focus on the use of Artificial Intelligence techniques to assess knowledge represented as causal maps.

In the map-evaluation process, myriad scoring systems that aim to evaluate students’ knowledge map consistently and accurately [[4](#_bookmark137), [49](#_bookmark182)] have been applied in research. These systems can be classified into two categories, with a few instances belonging to both (i.e., hybrid systems). *Reference-based scoring systems* compare a student’s map with an expert map. For example, if there are factors (i.e., nodes) present in the expert’s map but not in the student’s work, then they would be considered as missing. The reference map could be considered as the end goal: the student should eventually be able to replicate key features of that map to be deemed correct [[4](#_bookmark137)]. *Reference-free scoring systems* do not have such target maps; instead, a student’s map is assessed by itself, based on its structure and/or semantics. Such systems often treat and assess a knowledge map as a graph, where many components and structural measures are computed based on graph theory [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [10](#_bookmark143), [11](#_bookmark144)]. Studies on reference-free assessment have involved *many* measurement metrics (a.k.a, features) to eval- uate knowledge maps, thus reflecting the preferences of various research groups for different graph features. Considering that these systems are intended to support instructors in eval- uating their students’ work, the abundance of metrics means that instructors are expected to *select* and *interpret* these metrics themselves. *Selection* is a major barrier, as there is currently no evidence to identify which features *matter*. Instructors thus choose aspects of a map that they are more familiar with (e.g., number of nodes, edges, and density), which can ignore critical parts (e.g., diameter) and waste time by examining redundant features (e.g., the number of nodes and edges already determine the density). *Interpretation* is also a challenge: since we do not know which features are particularly deserving of attention, there is a paucity of guidelines on relating scores on certain features to the quality of a map. For example, a graph algorithm can provide an instructor with the diameter of the spanning tree for each student map, but the instructor does not necessarily know whether a higher or lower number means a better map. The closest guidelines originate from Novakian concept maps [[77](#_bookmark210), [78](#_bookmark211), [79](#_bookmark212), [80](#_bookmark213)], which are primarily hierarchical and hence forbid certain features found

in causal maps. Such forbidden features include cycles, which are important characteristics of complex problems yet are often missing in a map [[81](#_bookmark214), [82](#_bookmark215), [83](#_bookmark216)].

In this chapter, our main contribution pertains to the selection problem. Specifically, we use Artificial Intelligence as a new means of automatically selecting features to evaluate knowledge maps, thus operationalizing the identification of “features that matter”. This contribution is achieved by employing the machine learning approach of unsupervised fea- ture selection by applying eight algorithms of different types (e.g., Laplacian Score [[84](#_bookmark217)], SPEC [[85](#_bookmark218)], FSSEM [[86](#_bookmark219)]) to automatically identify unique subsets from twelve commonly used features. Our experimental approach is performed over 202 maps obtained across four studies, which allows us to examine how the identification of features depends on domains and settings.

The remainder of this chapter is organized as follows. In Section [3.2](#_bookmark51), we first show that the literature has used a variety of graph features to evaluate knowledge maps. Then, we provide a technical background on the approach of unsupervised feature selection used in this chapter. In Section [3.3](#_bookmark59), we detail our methods, including how we collected, pre-processed, and analyzed our data from four cohorts. Next, Section [3.4](#_bookmark67) presents our results, consisting of the rankings of feature indicating the importance of knowledge map metrics in various cohorts and based on different feature selection methods. Lastly, we discuss how our results provide a new AI perspective on the problem of knowledge map evaluation and contribute to guiding future studies that assess learning via knowledge maps.

## Background

### Core Concepts of Graph Theory in Map Assessments

In map assessments, a map is considered as a directed, labeled graph, which is defined as *G* = (*V, E*), where *V* is the set of nodes or “concepts”, each with a descriptive label; and *E* is a set of directed edges that represent causal relationships from one vertex onto another, typed as “+” or “−” depending on whether the source increases or decreases the target, respectively. When analyzing knowledge maps via reference-free scoring systems, different

measures are used for different evaluation purposes. These measures fall into two categories depending on the unit of analysis, which are summarized in Table [3.1](#_bookmark53).

*Graph-level* measures are computed over the whole graph, hence each such measure pro- duces one number. For example, the number of nodes, the number of edges, and the length of a map’s diameter are commonly selected to denote the problem space [[5](#_bookmark138)]. Thus, maps with more nodes and edges represent a broader scope of the problem. To capture the complexity of the problem, measures such as the number of cycles and chains are typically included [[5](#_bookmark138), [87](#_bookmark220)]. *Node-level* measures are performed for each node, thus delivering one number per node.

These measures aim to capture the “centrality” of each node, which may depend solely on its immediate neighborhood (e.g., degree centrality is the number of neighbors per node) or mobilize the broader graph based on network notions such as distances (e.g., eccentricity centrality, closeness centrality), control of flows (e.g., betweenness Centrality), or matrix

algebra (e.g., eigenvector centrality). Unlike graph-level measures, node-level measures are not primarily used to evaluate the quality of a map. So far, they have served to reveal focal points, or compare perspectives between maps [[11](#_bookmark144)]. Our focus in this chapter is thus on graph-level measures.

**Metrics Explanation Studies**

**Graph Level Measures for the whole graph**

Order Number of vertices/nodes. [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [12](#_bookmark145), [46](#_bookmark179), [88](#_bookmark221)]

Size Number of edges. [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [12](#_bookmark145), [46](#_bookmark179), [88](#_bookmark221)]

Average Degree Average number of edges per vertex [[4](#_bookmark137), [5](#_bookmark138)]

Density Number of edges existing as a ratio of maximum edges.

Connected There is a path from any node to any other node in a graph.

Diameter The furthest length between two vertices of the graph.

Cycles A cycle is a path that travels more than one node and ends at the starting point. Detailed metrics in- clude number of cycles and average length of cycles.

[[4](#_bookmark137), [12](#_bookmark145), [88](#_bookmark221)]

[[4](#_bookmark137)]

[[4](#_bookmark137), [12](#_bookmark145), [46](#_bookmark179), [88](#_bookmark221)]

[[4](#_bookmark137), [12](#_bookmark145), [38](#_bookmark171), [88](#_bookmark221)]

Number of paths Total number of existing paths. [[4](#_bookmark137), [5](#_bookmark138)]

Compactness Average path length [[4](#_bookmark137)]

Diameter of spanning tree Spanning tree is a portion of the graph where all

nodes are visited once and the visiting path is acyclic. The diameter is the shortest path between the two most distant nodes.

Ruggedness Number of components. Components are discon- nected parts of a graph.

**Vertex Level Centrality measures for a single vertex**

Closeness Centrality The closeness of a vertex describes how close a ver-

tex is to every other vertex. If the sum of the dis- tances is larger, then the closeness is smaller.

Betweenness Centrality The betweenness of a vertex represents the impor-

tance of a vertex within a graph. The betweenness of a specific vertex is equal to the number of a vertex that consists of a part in the shortest path between all the other vertices.

Eccentricity Centrality The eccentricity of a node is the maximum distance

from the node to all other nodes in a graph.

Eigenvector Centrality The eigenvector centrality measures the ease of ac-

cessibility of a node to all other nodes.

[[4](#_bookmark137), [7](#_bookmark140), [12](#_bookmark145)]

[[4](#_bookmark137), [46](#_bookmark179)]

[[4](#_bookmark137), [11](#_bookmark144)]

[[4](#_bookmark137), [11](#_bookmark144), [89](#_bookmark222)]

[[90](#_bookmark223)]

[[89](#_bookmark222), [90](#_bookmark223), [91](#_bookmark224)]

Table 3.1: A list of metrics that have been used in studies of knowledge maps within the context of network science.

### Unsupervised Feature Selection Methods

*Feature selection* is widely applied in areas such as machine learning, data mining, and statistical analysis. It is also one of the most powerful techniques for dimensionality reduc- tion, which aims to select the most relevant subset of features and remove irrelevant features from the input data [[92](#_bookmark225), [93](#_bookmark226), [94](#_bookmark227)]. Thus, the selected subset of features is able to present the vital features of the whole input data [[94](#_bookmark227), [95](#_bookmark228)]. According to the availability of labels in the data sets, feature selection methods can be classified as *supervised*, *semi-supervised*, and *unsupervised* [[96](#_bookmark229), [97](#_bookmark230)]. Supervised feature selection methods include well-known approaches, such as computing the Pearson’s correlation or Chi-square metric between the label and the numerical features. However, our problem context does not have such labels: we do not already know whether a map is “good” or “bad”, and such labels would anyway depend on how instructors prioritize certain features. Given the lack of labels, we focus on Unsuper- vised Feature Selection (UFS) methods. An example of UFS is illustrated in Figure [3.1](#_bookmark55). In the example, Figure [3.1](#_bookmark55)-a shows that feature f1 is able to discriminate clusters while f2 and f3 in Figures [3.1](#_bookmark55)-b and [3.1](#_bookmark55)-c do not add any significant information to the clustering; hence, the feature selection process suggests that f2 and f3 could be removed [[92](#_bookmark225)].

Based on the strategy used for selecting features, UFS fall into two categories (*filter* and *wrapper*), which can also be combined via hybrid approaches. Both categories are used in our work. We describe their main mechanisms and differences in the following subsections.

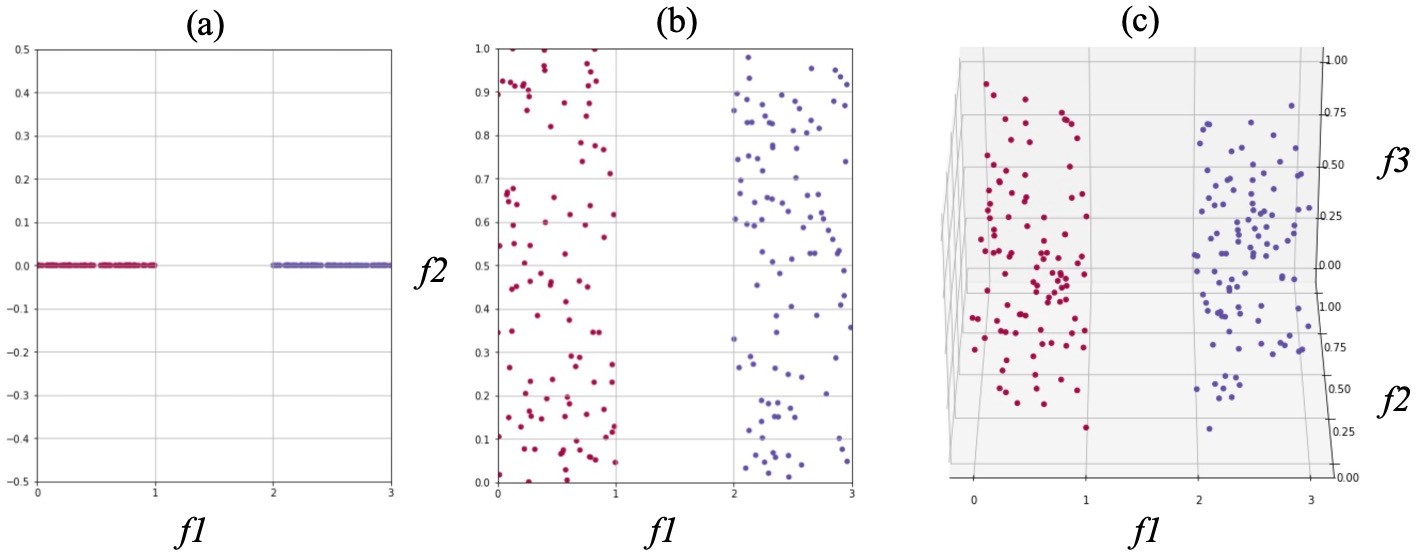


Figure 3.1: An example of unsupervised feature selection approach.

##### Filter Methods

Filter methods apply an evaluation criterion to assign a value to each feature, which can then be sorted from most to least important. In other words, filter methods select the most relevant features based on a performance measure by using intrinsic properties of the data. The main advantage of filter methods is their speed and the ability to rank

individual feature or various feature subsets [[98](#_bookmark231)], hence they are often used as a preliminary step to a learning algorithm (Figure [3.2](#_bookmark56)). Among filter methods, features commonly quantify similarities between entities using either *Information* Theory [[99](#_bookmark232)] or *Spectral* Analysis [[100](#_bookmark233), [97](#_bookmark230)].

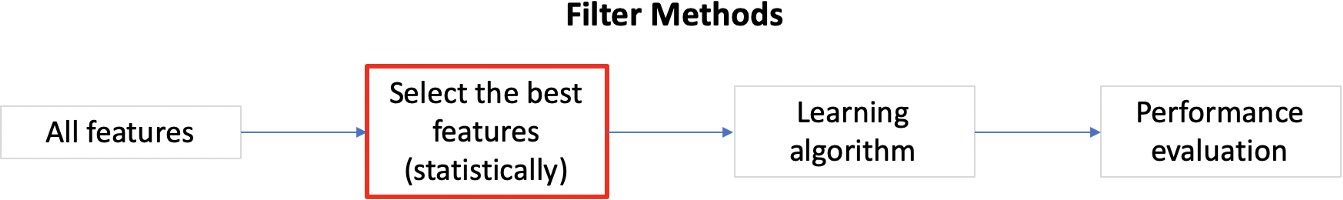


Figure 3.2: The Framework of filter feature selection methods.

One information-based UFS method is the Sequential backward selection method for Un- supervised Data, known as *SUD*, which uses entropy (originating from Information Theory) to evaluate the distance between pairs of objects [[101](#_bookmark234)]. If the entropy is high, the two ob- jects are far away; conversely, the lower the entropy, the more clustered the data. Another information-based UFS method is SVD-Entropy [[102](#_bookmark235)], which also measures the entropy of the original data, but through its singular values. Each feature is evaluated through a leave- one-out comparison, and its Contribution to the Entropy (CE) is documented. Selected features are sorted according to their CE accordingly [[102](#_bookmark235), [103](#_bookmark236)].

For spectral-similarity based UFS methods, the *LapScore* methods uses the Laplacian Score [[84](#_bookmark217)], which captures the importance of a feature by calculating its variance. A higher weight is assigned to features that are better able to preserve the local structure of the graph [[96](#_bookmark229), [97](#_bookmark230)]. LapScore can evaluate each feature individually, which enables it to rank the feature importance and select the top *k* features with smaller LapScore [[96](#_bookmark229)]. LapScore has been generalized through another spectral-similarity based method, known as *SPEC* (Spectrum decomposition) [[85](#_bookmark218)]. SPEC assigns similar values to features that behave similarly in the same class/cluster [[92](#_bookmark225), [85](#_bookmark218)]. This is accomplished by building a similarity matrix among the graphs, based on their structure. Two other spectral-similarity based UFS methods, such as *UDFS* (Unsupervised Discriminative Feature Selection algorithm) [[104](#_bookmark237)] and *NDFS* (Nonnegative Discriminative Feature Selection) [[105](#_bookmark238), [106](#_bookmark239)], involve Sparse Learning [[107](#_bookmark240)] to create a trade-off between goodness-of-fit and sparsity. That is, both UDFS and NDFS select a subset of discriminative features by simultaneously performing spectral clustering and applying a regularized regression model. The main difference between UDFS and NDFS is that NDFS incorporates non-negative constraints in the spectral clustering process, where pseudo class label (defined as non-negative) indicators can be obtained [[96](#_bookmark229), [97](#_bookmark230)].

##### Wrapper Methods

Wrapper methods select features based on the performance of a modeling algorithm such as clustering. The performance evaluation is a repetitive process which aims to finding the optimized subset of features that contribute to the best result in the model [[92](#_bookmark225), [97](#_bookmark230), [98](#_bookmark231)].

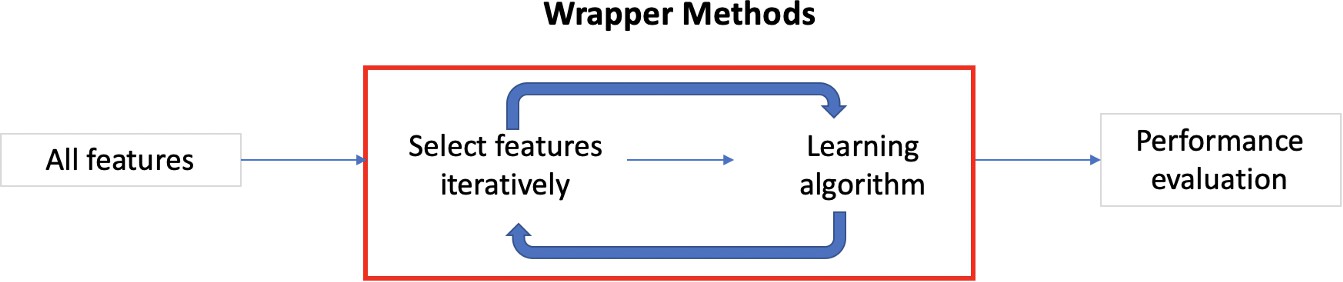


Figure 3.3: The Framework of wrapper feature selection methods.

One common strategy in wrapper methods is sequential methods, which add or remove feature sequentially. The selection strategy can be achieved through forward, backward, or bidirectional selection. Intuitively, a *forward* selection strategy means that wrapper methods select the one feature that has the best score (e.g., with respect to a clustering algorithm) and add it into the optimal subset, then re-compute the score for the remaining features to select the best one, and so on until the number of selected features meets a requirement [[92](#_bookmark225), [97](#_bookmark230), [98](#_bookmark231)]. The repeated nature of the process is illustrated in Figure [3.3](#_bookmark57). The score associated with each feature thus shows how good it is within that one round. For example, during a first round, “connected” may be the first feature selected among 12; in our case, with an associated score of 1*.*5*e*26. In the second round, “connected” is removed and the scores for the remaining features are re-computed, thus revealing the average path length as best feature with a score of 2*.*9*e*27. This example shows that features cannot be ranked by their best scores *across* rounds, as that would have erroneously resulted in considering the average path length to be better than connectedness. Rather, one feature is selected at a time, based on the best score in a given round. Conversely, a *backward* selection strategy starts with the whole set of features and gradually removes them.

In both cases (forward and backward strategies), ranking reflects the *order* in which features have been selected. That is, it depends on when they were added (for forward selection) or removed (for backward selection). The main disadvantage of wrapper methods is their eﬀiciency, as they are much slower than filters methods in searching for good subsets of features. However, they are empirically proven to have better performance in selecting features that subsequently improve classification or clustering [[92](#_bookmark225), [98](#_bookmark231)].

The algorithm for feature subset selection using Expectation Maximization (EM) clus- tering (known as *FSSEM* ) evaluates features via criteria such as scatter separability or maximum likelihood [[86](#_bookmark219)]. Another wrapper method that also adopts sequential searching strategy and scatters separability as the evaluation criteria is the Localized Feature Selec- tion Based on Scatter Separability, known as *LFSBSS* [[108](#_bookmark241)]. LFSBSS applies a sequential *backward* elimination process to select features, hence it first generates clusters by using the whole feature space and then removes irrelevant features iteratively based on scatter sepa- rability [[92](#_bookmark225), [108](#_bookmark241)]. Similarly to filter methods of UDFS and NDFS, the Dependence Guided Unsupervised Feature Selection (known as *DGUFS*) uses a constrained regression model to select features and produce clusters [[109](#_bookmark242)].

Among UFS wrapper methods, the *k*-means clustering algorithm has also been widely applied. In the k clusters, features minimizing the within-cluster distance and maximizing between-cluster distance are assigned a higher weight of importance [[92](#_bookmark225)]. Technically, *k*- means is a mixed methods combining ideas from wrappers and filters: a higher weight means that a feature is more important, hence features can be sorted (as in filter-based methods); however, the objective function depends on a controlling parameter *β* hence rankings are computed repeatedly (which echoes the iteration of wrapped-based methods), for different values of the parameter, to examine the stability of the ordering. A popular variation of *k*-means is the Entropy Weighting k-means (known as *EWKM* ), which includes the entropy in the objective function used by *k*-means to weight features [[110](#_bookmark243)]. Another variation on *k*-means variation is the Weighting K-Means (known as *W-k-means*), where features with larger weights are deemed more important [[111](#_bookmark244)].

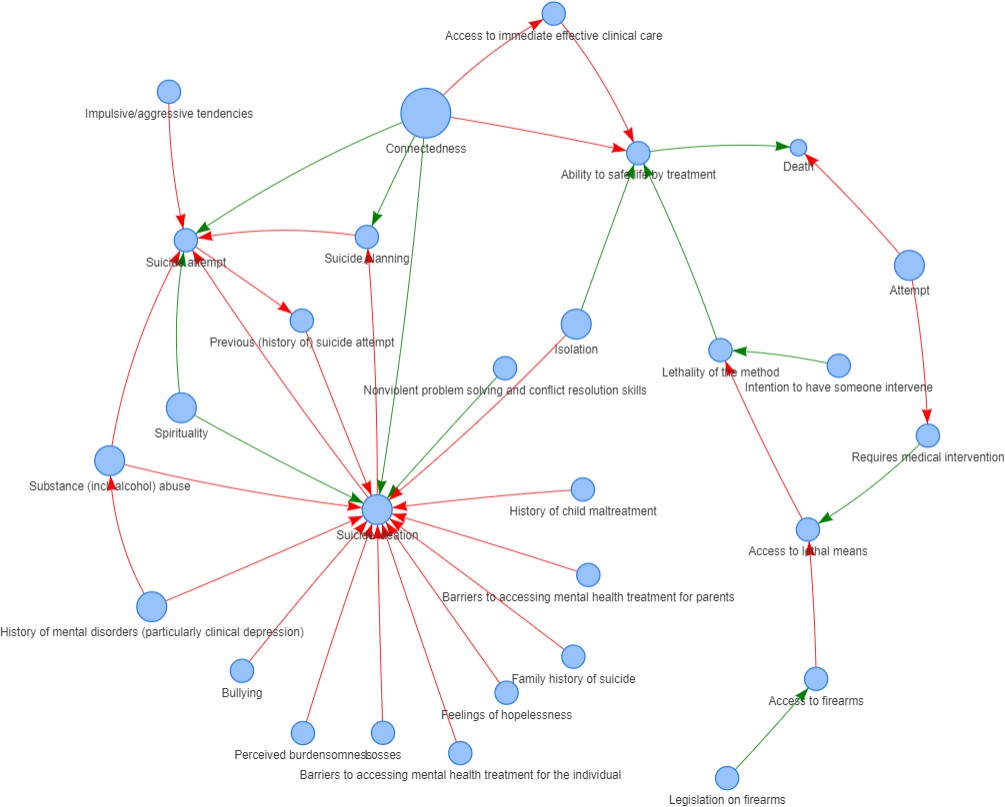


Figure 3.4: Sample expert map from Dataset A, created by a facilitator after a one-on-one interview was transcribed.

## Methods

### Data Collection

The knowledge maps in this chapter originate from four case studies, all of which have been published previously. These case studies cover a diversity of application domains (sui- cide, hiring, urban planning, obesity) and participant profiles, which helps to analyze how the importance of certain structural features in a map may depend on the context.

*Dataset* ***A*** originates from a survey of 16 experts in summer 2020 on the topic of youth suicide [[112](#_bookmark245)]; experts identified the risk and preventative factors for various stages involved in suicide (ideation, planning, attempt, death) as well as adverse childhood experiences. Each expert produced a map (hence 16 maps) as a result of a one-on-one semi-structured interview held virtually, via Zoom (Figure [3.4](#_bookmark58)). Since this dataset reflects experts rather than novice learners, it serves as a comparison to the other three datasets representing the more traditional learners that instructors would be evaluating.

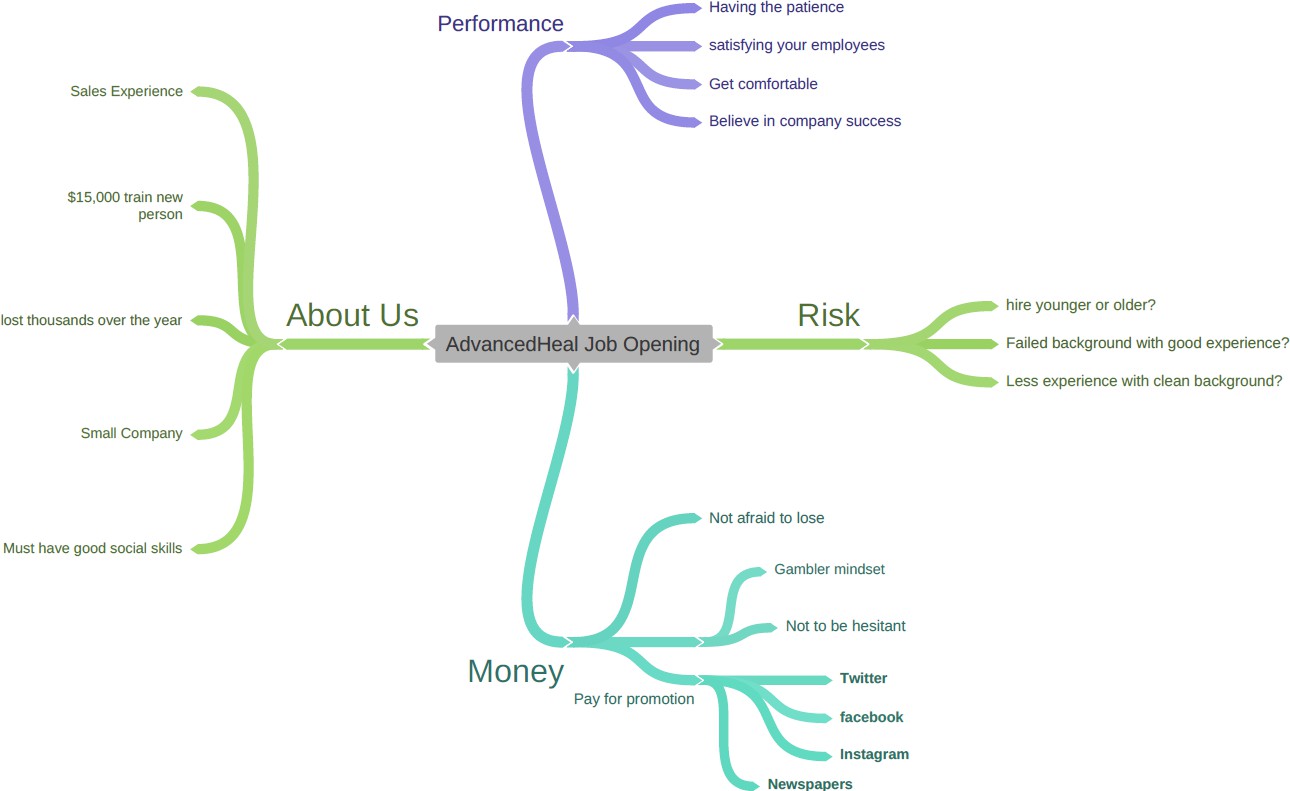


Figure 3.5: Sample student map from Dataset B, created by the students themselves via

*Coggle* for an assignment.

*Dataset* ***B*** was collected in summer 2021 from assignments completed by students in the Marketing Department at the University of Missouri Trulaske College of Business. Students did not receive extensive training on mapping prior to externalizing their views as maps. This resulted in 60 maps, on a case study examining whether an internal or external candidate should be hired, thus reflecting issues such as equity, company culture, and team motivation (Figure [3.5](#_bookmark61)). This case study has been used in numerous prior studies over ten years in

education research [[113](#_bookmark246), [114](#_bookmark247), [115](#_bookmark248)] and originates from an authentic decision-making situation conveyed by an expert [[116](#_bookmark249)].

*Dataset* ***C*** was collected in three semesters (Spring and Fall 2018, Spring 2019) at two other American institutions, Northern Illinois University in the Midwest and Furman Uni- versity in South Carolina. Students received extensive training on mapping prior to data collection by taking a course in Systems Thinking, in Network Science, or in Knowledge Representation and Reasoning. Students produced 105 maps regarding a case study arguing on the use of public subsidies to attract a sizable company, which would affect the econ- omy, local residents, and urban planning. Details about the data collection process and a comparison of the sub-groups are provided in [[5](#_bookmark138)].

Finally, *dataset* ***D*** collected 21 maps in Spring and Summer 2019 at Furman University on the topic of obesity. Each student was given a public health report, thus providing an evidence base on the problem. Students were then instructed to transform the report into a map, hence this dataset reflects their ability for transferring knowledge from a specific source rather than the more open-ended nature of the case studies in datasets B and C. The content of the map was examined and compared with the reports in a previous study [[88](#_bookmark221)].

### Data Pre-processing

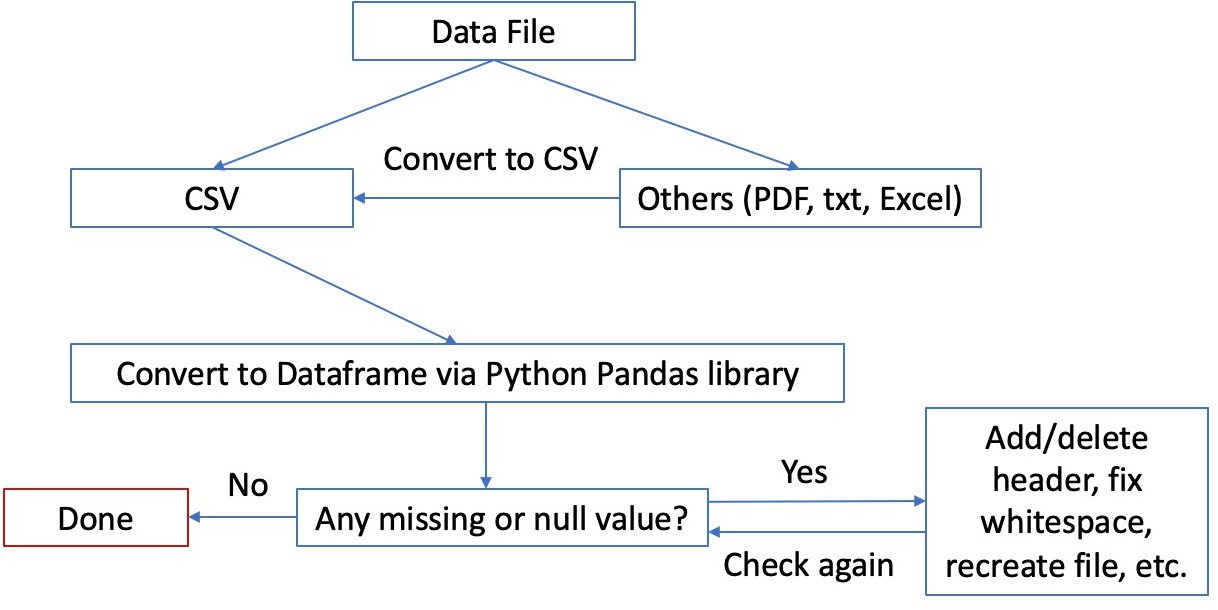


Figure 3.6: Overview of the data cleaning process.

The overall data pre-processing is illustrated in Figure [3.6](#_bookmark63). We first converted all 202 collected data files, including txt, pdf, csv, and excel formats, into a uniform CSV format for later analysis. Each CSV file is organized as a list of edges, that is, each row provides the starting node (1st column) and the end node (2nd column). Common issues are fixed

as part of this conversion, such as typos (e.g., extra whitespace) in the name of the nodes. For example, if there is a pair of edges “Stress → Depression” and another pair “Stress → Depresion”, this would erroneously create two distinct nodes (Depression, Depresion) hence

we resolve the typo. Another case would be the inclusion of a comma as part of a concept name, which causes errors since CSV are comma-separated files; commas are thus removed from node labels. Additional errors are caught when we load a CSV file into a Dataframe via Python Pandas library [[117](#_bookmark250)]; such cases are fixed manually to ensure that data files are read and analyzed in the correct graph structure.

### Data Analysis

After the maps for each of the four studies are pre-processed, we perform two steps for the analysis. First, we *extract features*, that is, we compute each of the graph metrics considered in the prior literature for each of the maps. This generates a dataset of measurements, which is then suitable for a machine learning process. Second, we *select features* by using several unsupervised feature selection (UFS) algorithms on the dataset of measurements. Each UFS algorithm provides a ranking of features, which it considers most informative to characterize a given dataset. Since the UFS algorithms reflect different strategies (Section [3.2.2](#_bookmark54)), the rankings can be different and thus have the possibility of supporting diverse instructor pro- files. The rankings are presented in Section [3.4](#_bookmark67). The algorithms for data analysis are openly available as a Jupyter Notebook on a third-party repository by the Open Science Framework at <https://osf.io/rneku/>.

**Metrics Implementation**

Number of nodes (a.k.a Order)

Number of edges (a.k.a Size)

Counted the number of nodes |*V* (*G*)| in a graph G via NetworkX’s num- ber\_of\_nodes() function.

Counted the number of edges |*E*(*G*)| in a graph G via NetworkX’s num- ber\_of\_edges() function.

Average Degree Found the value of average degree by |*E*(*G*)| .

|*V* (*G*)|

Density Found the value of density by 2|*E*(*G*)| .

|*V* (*G*)|∗(|*V* (*G*)|−1)

Connected Found whether a graph G is connected via NetworkX’s is\_connected(G) function. Diameter Found the longest of the shortest paths in a graph G via NetworkX’s short-

est\_simple\_paths(G) function.

Number of cycles Found the number of cycles in a graph G via counting the number of the cycle list returned by NetworkX’s cycle\_basis(G) function.

Average cycle length Found the average length of cycles in a graph G via getting the mean length from the cycle list returned by NetworkX’s cycle\_basis(G) function.

Number of path Found by getting all shortest paths between nodes in a graph G via NetworkX’s shortest\_simple\_paths(G) function.

Average path length (a.k.a Compactness) Diameter of spanning tree

Number of components (a.k.a Ruggedness)

Found by getting the mean of all shortest paths between nodes in a graph G via NetworkX’s shortest\_simple\_paths(G) function.

Found by getting the diameter of the minimum spanning tree SPTree in a graph G via NetworkX’s diameter(minimum\_spanning\_tree(G)) function.

Found by getting the connected components in a graph G via NetworkX’s num- ber\_connected\_components(G) function.

Table 3.2: Metrics for data analysis along with how they are implemented.

##### Feature Extraction via Graph Metrics

As discussed in Section [3.2.1](#_bookmark52), a graph can be analyzed at two levels: as a whole (graph- level) or via its individual nodes (node-level). To assess a map, an analysis is performed at the graph level. The motivation for our work is that there are many graph-level metrics to consider, hence we implemented all metrics identified in a scoping review of the litera- ture on map assessment (Table [3.1](#_bookmark53)). Our implementation relies on the NetworkX Python library [[118](#_bookmark251)], which already provides suﬀicient functions to perform several of these metrics (Table [3.2](#_bookmark65)).

##### Unsupervised Feature Selection

We selected the optimal subset of metrics from the collected maps by considering eight different UFS algorithms, including five filter methods and three wrapper methods. The im- plementations of these eight UFS primarily relied on the *scikit-feature* Python library [[96](#_bookmark229)], as well as our own implementations for other algorithms based on their publications (Table [3.3](#_bookmark66)).

**Algorithms Approach Output Refs. Parameters**

LapScore Filter Rank/select features based on Lap-

Scores in descending order.

SPEC Filter Rank/select features based on

SPEC values in descending order. UDFS Filter Rank/select features based on fea-

ture weights in descending order. NDFS Filter Rank/select features based on fea-

ture weights in descending order. SVD-Entropy Filter Rank/select features based on en-

tropy contribution of the original data matrix through its singular values in descending order.

FSSEM Wrapper Rank/select features based on Scat-

ter Separability Criterion or Max- imum Likelihood Criterion in de- scending order.

W-k-means Wrapper Rank/select features based on the

weight of feature importance in de- scending order.

DGUFS Wrapper Select subset of features in the op-

[[84](#_bookmark217), [96](#_bookmark229)] metric: euclidean

[[96](#_bookmark229), [85](#_bookmark218)] metric: euclidean

[[96](#_bookmark229), [104](#_bookmark237)] n\_cluster: 4; k: 5

[[96](#_bookmark229), [106](#_bookmark239)] n\_cluster: 4

[[102](#_bookmark235)] N/A

[[86](#_bookmark219)] n\_cluster: 4

[[111](#_bookmark244)] n\_cluster: 4

[[109](#_bookmark242)] n\_cluster: 4

timization process.

Table 3.3: Eight selected UFS algorithms and what they output, along with their implemen- tations.

Several of the algorithms have parameters. In order to have a fair comparison of their rankings, we set their parameters on the same target and to match the behavior of the other algorithms. For example, we set the parameter *k* to 5 in the UDFS to govern its use of the *k*-nearest neighbors algorithm [[119](#_bookmark252)], thus matching the default neighborhood size in other

algorithms including LapScore, SPEC, and NDFS. Similarly, we set them onto four clusters. If any of wrapper algorithm asks for the indicated number of features to select, we perform a parameter sweep by varying the numbers of selected features from 2 to 12 (which is the maximum number of features) to check the selection order and the consistency of selected subset of features. For algorithms that incorporate *K*-means clustering, we search for the best *K* via inertia score, which measures how well a data set was clustered by *K*-Means. Then, we output the rank or selected subset of metrics for all algorithms.

## Results

**Data Name # nodes # edges**

**Sets**

**avg density**

**degree**

**connected (1=yes, 0=no)**

**diameter # cycles**

**avg length cycles**

**# compo- nents**

**diameter spanning tree**

**# paths**

**avg path length**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **mean** | 27.22 | 35.47 | 1.36 | 0.16 | 0.89 | 5.80 | 9.72 | 3.10 | 1.47 | 6.16 | 473.86 | 2.91 |
| **All Data Sets (n=202)** | **std min 25%**  **50%**  **75%** | 18.83  5.00  15.00  21.00  33.00 | 25.04  5.00  20.00  27.00  40.75 | 0.44  0.72  0.97  1.28  1.63 | 0.12  0.02  0.06  0.13  0.24 | 0.32  0.00  1.00  1.00  1.00 | 2.47  2.00  4.00  6.00  7.00 | 12.22  0.00  0.25  6.00  14.00 | 2.12  0.00  0.25  3.60  4.16 | 1.62  1.00  1.00  1.00  1.00 | 3.31  0.00  5.00  6.00  8.00 | 748.82  10.00  91.00  183.50  435.00 | 1.00  1.33  2.10  2.79  3.45 |
|  | **max** | 100.00 | 126.00 | 2.83 | 0.67 | 1.00 | 16.00 | 64.00 | 12.00 | 13.00 | 18.00 | 4950.00 | 6.32 |
|  | **mean** | 61.19 | 83.00 | 1.36 | 0.05 | 0.88 | 7.94 | 23.00 | 4.96 | 1.19 | 9.00 | 2005.75 | 3.72 |
| **Data Set A**  **(n=16)** | **std min 25%**  **50%**  **75%** | 20.06  29.00  48.00  54.50  69.50 | 28.12  38.00  61.50  86.50  109.50 | 0.21  1.05  1.24  1.33  1.42 | 0.02  0.02  0.04  0.05  0.06 | 0.34  0.00  1.00  1.00  1.00 | 2.14  5.00  6.00  8.00  9.00 | 14.00  4.00  11.75  23.00  29.00 | 1.23  3.33  4.14  4.76  5.84 | 0.54  1.00  1.00  1.00  1.00 | 4.07  0.00  8.00  9.50  11.25 | 1346.87  406.00  1128.00  1459.00  2410.75 | 0.77  2.73  3.19  3.74  3.95 |
|  | **max** | 100.00 | 126.00 | 1.94 | 0.09 | 1.00 | 14.00 | 60.00 | 7.56 | 3.00 | 14.00 | 4950.00 | 5.91 |
|  | **mean** | 27.80 | 28.13 | 1.00 | 0.09 | 1.00 | 7.22 | 1.33 | 1.43 | 1.00 | 7.42 | 470.20 | 3.79 |
| **Data Set B**  **(n=60)** | **std min 25%**  **50%**  **75%** | 14.10  11.00  17.00  25.00  32.00 | 15.57  10.00  16.75  24.50  32.25 | 0.10  0.91  0.94  0.96  1.00 | 0.04  0.03  0.06  0.08  0.12 | 0.00  1.00  1.00  1.00  1.00 | 2.79  3.00  5.75  7.00  9.00 | 3.05  0.00  0.00  0.00  1.00 | 2.52  0.00  0.00  0.00  3.53 | 0.00  1.00  1.00  1.00  1.00 | 2.99  3.00  5.75  7.00  9.00 | 544.59  55.00  136.00  300.00  496.00 | 1.02  2.07  2.94  3.56  4.24 |
|  | **max** | 76.00 | 87.00 | 1.53 | 0.19 | 1.00 | 16.00 | 17.00 | 12.00 | 1.00 | 18.00 | 2850.00 | 6.32 |
|  | **mean** | 17.53 | 27.19 | 1.56 | 0.23 | 0.89 | 4.80 | 11.06 | 3.81 | 1.40 | 5.58 | 150.89 | 2.35 |
| **Data Set C**  **(n=105)** | **std min 25%**  **50%**  **75%** | 8.22  5.00  13.00  15.00  21.00 | 14.01  5.00  18.00  24.00  33.00 | 0.45  0.72  1.28  1.50  1.85 | 0.12  0.04  0.15  0.22  0.28 | 0.32  0.00  1.00  1.00  1.00 | 1.85  2.00  4.00  4.00  6.00 | 9.07  0.00  5.00  9.00  15.00 | 1.28  0.00  3.36  3.68  4.33 | 1.36  1.00  1.00  1.00  1.00 | 2.79  0.00  5.00  6.00  7.00 | 158.76  10.00  66.00  105.00  171.00 | 0.58  1.33  1.91  2.17  2.77 |
|  | **max** | 56.00 | 76.00 | 2.83 | 0.67 | 1.00 | 11.00 | 45.00 | 10.00 | 9.00 | 13.00 | 1285.00 | 4.05 |
|  | **mean** | 48.10 | 61.62 | 1.36 | 0.08 | 0.57 | 5.14 | 16.90 | 2.87 | 3.38 | 3.33 | 932.00 | 2.61 |
| **Data Set D**  **(n=21)** | **std min 25%**  **50%**  **75%** | 21.19  12.00  33.00  53.00  59.00 | 30.40  21.00  33.00  54.00  82.00 | 0.52  0.85  0.94  1.03  1.75 | 0.08  0.02  0.03  0.06  0.08 | 0.51  0.00  0.00  1.00  1.00 | 1.24  3.00  4.00  5.00  6.00 | 22.00  0.00  1.00  10.00  25.00 | 1.66  0.00  3.11  3.76  3.86 | 3.44  1.00  1.00  1.00  5.00 | 3.10  0.00  0.00  4.00  6.00 | 789.59  66.00  357.00  528.00  1378.00 | 0.40  1.83  2.40  2.63  2.83 |
|  | **max** | 94.00 | 116.00 | 2.19 | 0.32 | 1.00 | 8.00 | 64.00 | 4.20 | 13.00 | 8.00 | 2779.00 | 3.23 |

Table 3.4: Metrics distribution across all five data sets of collected maps.

For each of the graph metrics in Table [3.2](#_bookmark65), Table [3.4](#_bookmark68) summarizes the characteristics of each dataset as well as a pooled dataset in terms of central tendency, dispersion, and distribution. Figure [3.7](#_bookmark69) presents the feature selection results generated from eight different UFS algorithms for the pooled dataset, thus examining the *general* question of selecting features to evaluate a map. The features are displayed in a descending order based on the different evaluation criteria specified in each algorithm. A higher rank (i.e., features at the top of the table) indicates a higher importance or more relevancy in the selection process of

an algorithm. In order to nuance the analysis based on the characteristics of each dataset, we also provide the sum and average rank of each feature per dataset in Table [3.5](#_bookmark70). Note that a lower rank is better, that is, 1 is the best feature and 12 is the least.

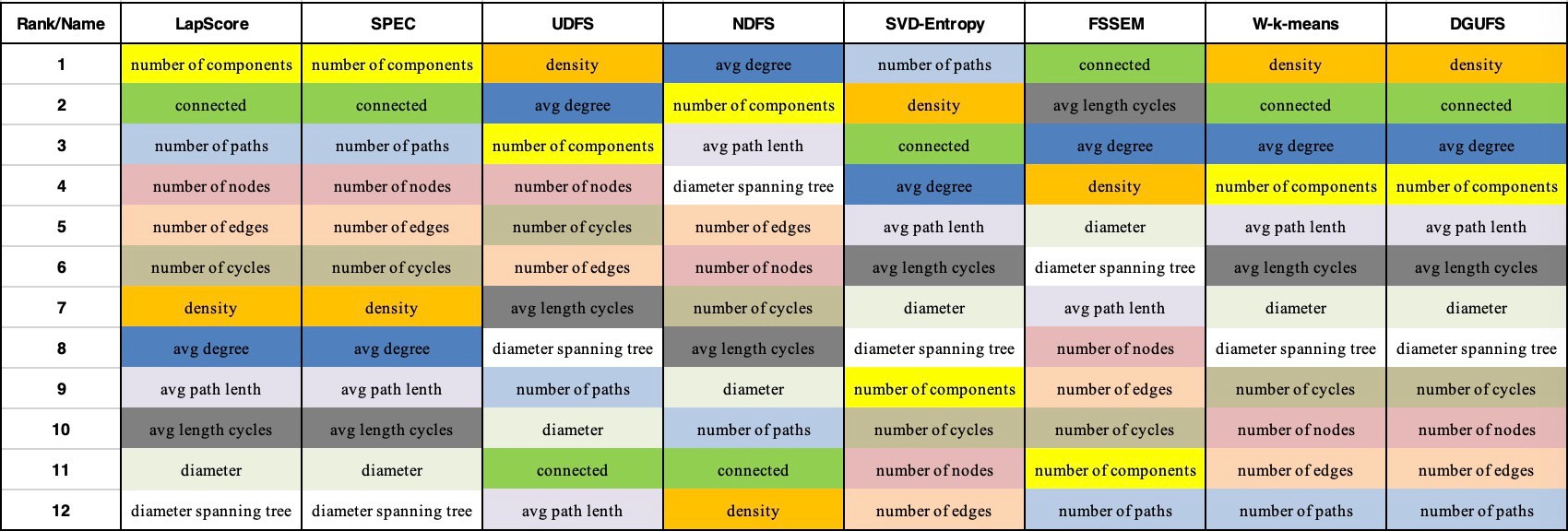


Figure 3.7: Results of feature rank/selection from eight different UFS algorithms are dis- played in a descending order based on the importance. Same features are highlighted in same color.

Based on the pooled results from all 202 knowledge maps, the top four important fea- tures/metrics are the average degree, connectedness, density, and number of components. The three least important features/metrics in this pooled analysis are the number of edges, diameter of spanning tree, and diameter. The fact that connectedness and the number of components are selected as top features may stem from how selection algorithms emphasize the structure of maps. That is, these methods assign high weights to features that most preserve the graph structure (manifold structure), commonly represented by the Laplacian matrix [[97](#_bookmark230)].

The analysis conducted at the level of each dataset helps us to (i) identify findings that may be generalizable and (ii) isolate effects where characteristics of the learners or tasks lead to different rankings (Table [3.5](#_bookmark70)). Across datasets, we see that three features are consistently deemed less informative (as indicated by the high rank): the number of paths, number of edges, and average path length. The connectedness and density of maps are considered potentially informative across all datasets. When examining the unique effects of each dataset, we make three observations. First, the average degree is not a useful feature to characterize expert maps, but it is essential for the maps of learners, being the best ranked feature in two cases (datasets B and D) and the fourth one in dataset C. Second, the diameter is not generally useful except for dataset D, which is the only case in which the map was obtained from a report instead of being elicited from the participants’ own views. Third, although connectedness is generally important because selection algorithms emphasize the structure of maps, its relative importance takes into account the characteristics of the dataset. For example, in dataset B, *every* map is connected hence connectedness has a lower importance as a discriminative feature. In contrast, in datasets A and C, we expect *most*

**All maps (n=202)**

**Data Set A (n=16)**

**Data Set B (n=60)**

**Data Set C (n=105)**

**Data Set D (n=21)**

**Feature Total**

**Rank**

**Avg Rank**

**Total Rank**

**Avg Rank**

**Total Rank**

**Avg Rank**

**Total Rank**

**Avg Rank**

**Total Rank**

**Avg Rank**

**avg degree** 32 4 56 7 39 4.875 43 5.375 29 3.625

**connected** 34 4.25 36 4.5 40 5 26 3.25 44 5.5

**density** 35 4.375 37 4.625 44 5.5 35 4.375 33 4.125

**# components** 35 4.375 38 4.75 46 5.75 27 3.375 48 6

**avg length cycles**

**avg path length**

55 6.875 56 7 39 4.875 64 8 43 5.375

55 6.875 56 7 52 6.5 57 7.125 66 8.25

**# nodes** 57 7.125 53 6.625 56 7 60 7.5 66 8.25

**# cycles** 62 7.75 57 7.125 49 6.125 63 7.875 71 8.875

**# paths** 62 7.75 58 7.25 56 7 58 7.25 64 8

**# edges** 64 8 64 8 64 8 63 7.875 64 8

**diameter spanning tree**

66 8.25 55 6.875 76 9.5 70 8.75 52 6.5

**diameter** 67 8.375 58 7.25 63 7.875 58 7.25 44 5.5

Table 3.5: Sum of rank and average rank of each feature in the result of total eight UFS algorithms across different data sets.

maps to be connected (as shown by the high average), but a few are disconnected hence they stand out and connectedness is the number one feature. In dataset D, there is no clear expectation about connectedness (as the average is close to an “in-between” value of 0.5 and has a large dispersion), thus connectedness is less important.

Finally, we found that the ranking of features is influenced by the selection approach, which can be useful to reflect the various preferences of instructors. That is, selection mechanisms with a similar heuristic produce identical or highly similar outcomes, whereas mechanisms grounded in a different approach yield different rankings. As seen in Figure [3.7](#_bookmark69), LapScore and SPEC produced identical feature ranking because both of them used Laplacian score or its variant to weigh the features; UDFS and NDFS generated similar results as they involve a similar discriminative feature selection process; and W-k-means and DGUFS, two iterative wrapper methods that consider the selection as an estimation problem [[97](#_bookmark230)], also yield similar results. The last two algorithms produced distinct rankings due to their unique approaches, with SVD relying on an information-based approach and FSSEM being a sequential wrapper. Although we acknowledge the nuances produced by various UFS algorithms, the results suggest that the eight algorithms result reflect five distinct approaches when applied to causal maps.

## Discussion

Through knowledge maps, learners are able to elicit and externalize their knowledge structures into maps, which can be assessed by instructors to evaluate learners’ understanding of a domain (i.e., summative feedback) and/or offer guidance so that learners can bridge their knowledge gap (i.e., formative feedback) [[3](#_bookmark136), [4](#_bookmark137)]. Knowledge maps are widely applied to represent and evaluate the perspectives of students across several domains, with a wealth of studies showing the benefits of constructing and assessing such maps [[1](#_bookmark134), [2](#_bookmark135), [3](#_bookmark136), [4](#_bookmark137), [37](#_bookmark170), [39](#_bookmark172), [40](#_bookmark173)]. However, evaluating a map is a challenge for instructors, as there is little guidance on which features should be examined. Our study in this chapter aimed to bridge this gap by employing artificial intelligence to identify informative features, that is, aspects of a map that can be focal points for evaluation. We adopted eight Unsupervised Feature Selection (UFS) techniques, whose different selection algorithms can reflect the possibility that instructors value different aspects of a map. Our analysis was performed over 202 maps originating from four studies, which allows to identify features that are often prominent (regardless of the task or nature of learners) and those that are more context-dependent.

Our results shed light on how metrics could be selected when evaluating knowledge maps. *Generally*, the average degree, connectedness, density, and number of components are useful discriminating features. Interestingly, the widely reported metric of “number of edges” was not as informative, which suggests a necessity to change the predominant approaches [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [12](#_bookmark145), [46](#_bookmark179), [88](#_bookmark221)] that report the number of edges, nodes, and density – the density alone may suﬀice. Characteristics of the participants and the mapping tasks also have an influence, so it is useful for instructors to account for these *contextual factors* when choosing features for assessment. For example, if participants are close to the level of experts then the average degree may carry little value, whereas it is informative for novice learners. Despite its importance in several studies [[4](#_bookmark137), [12](#_bookmark145), [46](#_bookmark179), [88](#_bookmark221)], the diameter is not an essential factor when maps are produced based on the learners’ perspectives, but it is informative when learners act as “translators” by producing a map from a detailed case study.

For educators who are interested in incorporating knowledge maps into the evaluation of students’ learning, our study in this chapter provides guidelines to decide which map features to select as the evaluation criteria. Our process is also repeatable, hence researchers who have to evaluate a large number of maps (e.g., within the context of a MOOC) can reuse our approach to automatically identify salient features. Sharing our code on a third-party repository (<https://osf.io/rneku/>) contributes to this effort.

Our study in this chapter has two main limitations. First, all datasets were collected in the USA from college students or experts. Although diversity was obtained by involving three universities (Furman University, Northern Illinois University, and University of Missouri) and different case studies, our findings should not be generalized to K–12. Second, in an applied machine learning study, there is always another algorithm that could be used. While we used eight algorithms to cover two broad categories (filter, wrapper) and showed that they provide five distinct perspectives, it is possible that another algorithm would yield its own ranking.

The observation that algorithms provide different rankings is particularly interesting for

future work. Indeed, our aim was not to assume that instructors are identical and will evaluate their students through a fully standardized process. Rather, our process acts as a “sieve” by distinguishing between useful and less informative factors, while noticing that nuances in the ranking depend on the heuristics. Ultimately, instructors have their own preferences and thus the heuristic used by one may be different for another. Future studies should identify how preferences from instructors can be elicited and mapped to a ranking system, thus making a clear connection between what the instructor wishes to see in the students’ work and the features that should then be extracted for assessment.

## Conclusion

Our work is the first to adopt unsupervised feature selection (UFS) methods to identify informative graph features for knowledge map assessment. Our evaluation spans 202 maps from four different case studies, using eight UFS methods. Results suggest that some of the metrics commonly reported (e.g., number of edges) may not be as important as previously thought of assessment, thus showing potential alternatives depending on the context afforded by the types of participants and the learning tasks.

**Chapter 4**

# Transitioning from Individuals to Groups in Knowledge Map Construction

Unlike problems with few solutions (e.g., multiple-choice questions), in problem-based learning (PBL), students are facing complex problems that may admit multiple solutions. Students are expected to elicit and apply their knowledge in various ways beyond outputting the prescribed solutions. Knowledge map is an effective tool to support students in PBL, to represent, analyze, and guide a students’ knowledge structures. Previous studies have examined the use of knowledge mapping techniques in both individual and collaborative learning process. These studies focused on exploring the degree to which knowledge map- ping could influence students’ learning outcomes and their knowledge understanding, but they did not yet document how the map *structure* is altered in the transition from individual to collaborative mapping process. In this work-in-process study, we examined how 12 net- work science metrics (e.g., number of nodes, diameter) change when we transition from the individual maps of learners to the maps produced by groups. Based on 44 individual maps and their corresponding 10 group maps, our work-in-progress work suggests that 1) several map metrics that commonly denote the breadth (e.g., number of edges) and complexity (e.g., number of cycles) of problems significantly differ between individual and group maps;

2) smaller group size may have a slightly bigger influence on the change of map metrics in terms of metrics quantity, but similar group sizes may not noticeably impact the map.

## Introduction and Literature Review

Problem-based learning (PBL) approaches have been widely applied in multiple disci- plines as an instructional strategy [[120](#_bookmark253), [121](#_bookmark254)]. In PBL, students are presented with realistic, ill-structured, decision-making cases which admit multiple solutions [[120](#_bookmark253), [122](#_bookmark255)]. These prob- lems are designed to be representative of the type of issues often faced by domain practi- tioners. To solve such problems, learners are encouraged to integrate their understanding of theory and skills into practice [[121](#_bookmark254)] as well as carefully analyze the problem space [[122](#_bookmark255)]. For example, a case prompted students to decide when a new company should be granted local tax incentives [[5](#_bookmark138)]. Through problem-based learning curricula and the learning cycle (Figure [4.1](#_bookmark75)), learners working on such a case are expected to improve skills such as ability to construct a flexible knowledge base, develop effective problem-solving skills, acquire lifelong learning skills [[120](#_bookmark253)], and increase their critical thinking [[123](#_bookmark256)].

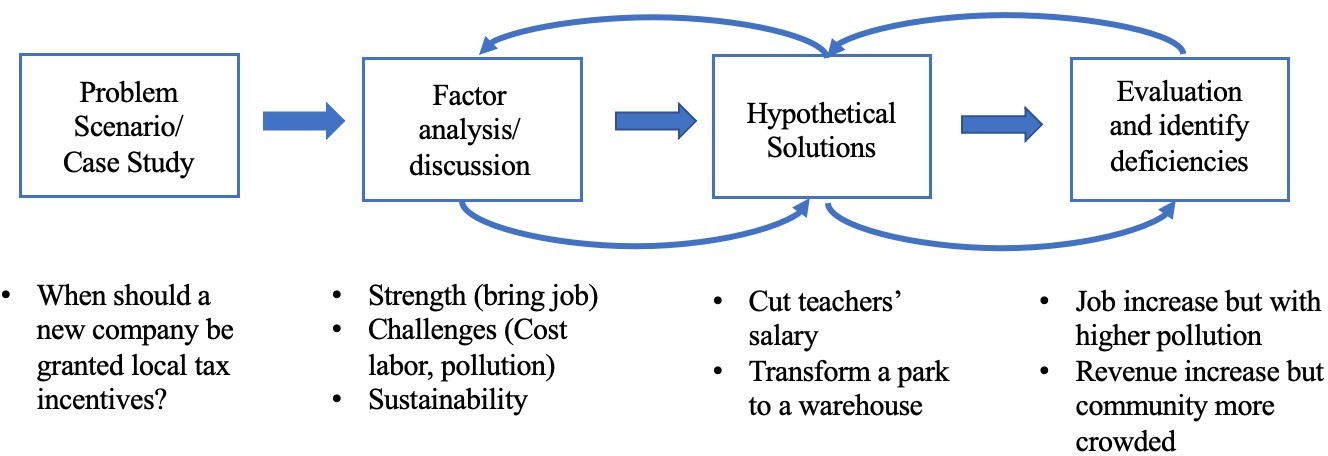


Figure 4.1: A learning cycle presents how students carried out a case study via the problem- based approach. This figure is modified from the learning cycle [[120](#_bookmark253)] by applying it on the case from [[5](#_bookmark138)].

As educators have explored ways to better support complex problem-solving, theorists assert that learners should be able to articulate their knowledge in various ways beyond prescribed correct answers [[124](#_bookmark257), [125](#_bookmark258)]. Indeed, learners in PBL should be afforded ways to represent knowledge not only to support their proposed resolution, but also to consider other solution paths that may serve as viable pathways. One way includes knowledge maps that can be used to visually depict learners’ structural knowledge. This approach allows learners to identify a series of concepts in the forms of nodes and describe the relationships between these concepts as links. The links can be labeled, which further allows them to characterize the relationships. Finally, learners can visually organize concepts, for example, by forming clusters to show proximity around distinct, yet similar concepts. In doing so, practitioners argue that the knowledge representation depicted in a knowledge map can be used as a proxy for mental model formation [[55](#_bookmark188), [125](#_bookmark258), [126](#_bookmark259), [127](#_bookmark260)].

Various studies have explored the use of knowledge mapping as part of the learning pro- cess. Early studies regarding knowledge maps often focused on the benefits of individual

knowledge maps, with evidence suggesting benefits in terms of conceptual change [[128](#_bookmark261)] and connected understanding among concepts [[129](#_bookmark262)]. As discourse around the benefits of peer interaction during problem-solving emerged, other studies began to explore the role of *col- laborative knowledge maps* on learning outcomes. For example, Kwon and Cifuentes [[130](#_bookmark263)] compared the effects of individually-constructed and collaboratively-constructed knowledge maps on conceptual understanding. They found that learners engaged in the collaborative component outperformed individuals on posttest scores. Further analysis of knowledge maps produced by groups revealed stronger quality maps that were reflective of deeper domain un- derstanding. Other case studies describe how collaborative knowledge mapping results in distinct higher order thinking skills, including elaboration [[131](#_bookmark264)], clinical reasoning [[132](#_bookmark265)], and meta cognition [[133](#_bookmark266)]. Large scale meta-analyses find evidence of advanced thinking in terms of both memory retention [[134](#_bookmark267)] and critical thinking [[135](#_bookmark268)].

## Research Questions

While these studies have explored how the mapping process impacts learners’ learning outcomes and their knowledge changes, these studies did not investigate how specific aspects of a map (e.g., number of nodes and edges, diameter) are altered when transitioning from an individual mapping process to group-model building. To bridge this research gap, this chapter examines the following research questions via a quasi-experimental design:

(Q1) Which aspects in terms of map structure are impacted by a transition from indi- vidual to group mapping?

(Q2) Is the effect mediated by the size of a group?

The remainder of this chapter is organized as follows. In Section [4.3](#_bookmark77), we provide detailed information on our participants, research design, procedures of data collection and data analysis on 12 metrics in maps produced by individuals and groups of different sizes. Next, we present our results, together with statistical tests and similarity analyses of maps in Section [4.4](#_bookmark86). Lastly, we discuss the significance of these results in terms of collaborative mapping in the learning process and highlight limitations pertaining to the work-in-progress nature of this study in Section [4.5](#_bookmark94).

## Methods

### Data Collection

Participants were enrolled in two sections of a course entitled “Principles of Sales Manage- ment”. The course was offered in the Marketing Department at the University of Missouri Trulaske College of Business, which is a large, research focused institution located in the Midwest portion of the United States. The course was conducted online, with dedicated dis- cussion board spaces in the learning management to support groups in collaborating. The inquiry-based module was amended as part of an existing course activity. Prior to course activity, participants completed the IRB consent form. Students did not receive extensive

training on mapping prior to performing the activity, hence the maps that they produced via *Coggle* (a mapping software) reflect their natural organization instead of following specific practices such as performing network analysis on one’s own map to improve it [[5](#_bookmark138)].

The case study examined whether an internal or external candidate should be hired to a position. This is a relevant problem to students in business, and it evokes multiple issues such as equity, fit with company culture, team motivation, or the uptake of new practices. Our team has used this case study, named “Nick’s dilemma”, for over 10 years in educational research to examine various aspects of mapping and collaboration among learners. For instance, we examined how individuals would change their decisions when exposed to positive or negative cases [[114](#_bookmark247)], used a recommender system to automatically find relevant cases [[136](#_bookmark269)], or would construct causal maps in response to the problem [[113](#_bookmark246)].

### Data Pre-processing

In examining the collected data, we found three submissions without identified authors, and one group did not complete the work as instructed. Therefore, we removed those maps from the data set, then we converted the remaining 54 collected maps (in formats such as txt and pdf) into a consistent csv format for the data analysis. These csv files encode the content as a list of edges, where each edge is directed by specifying a starting node and an end node. These lists of edges were reviewed using the Pandas Library [[117](#_bookmark250)] and NetworkX Python Library [[118](#_bookmark251)] to identify and remedy formatting issues, such as extra whitespace, headers, rows, columns, and combined content that cause missing or error values. The overall data pre-processing is illustrated in Figure [4.2](#_bookmark80)

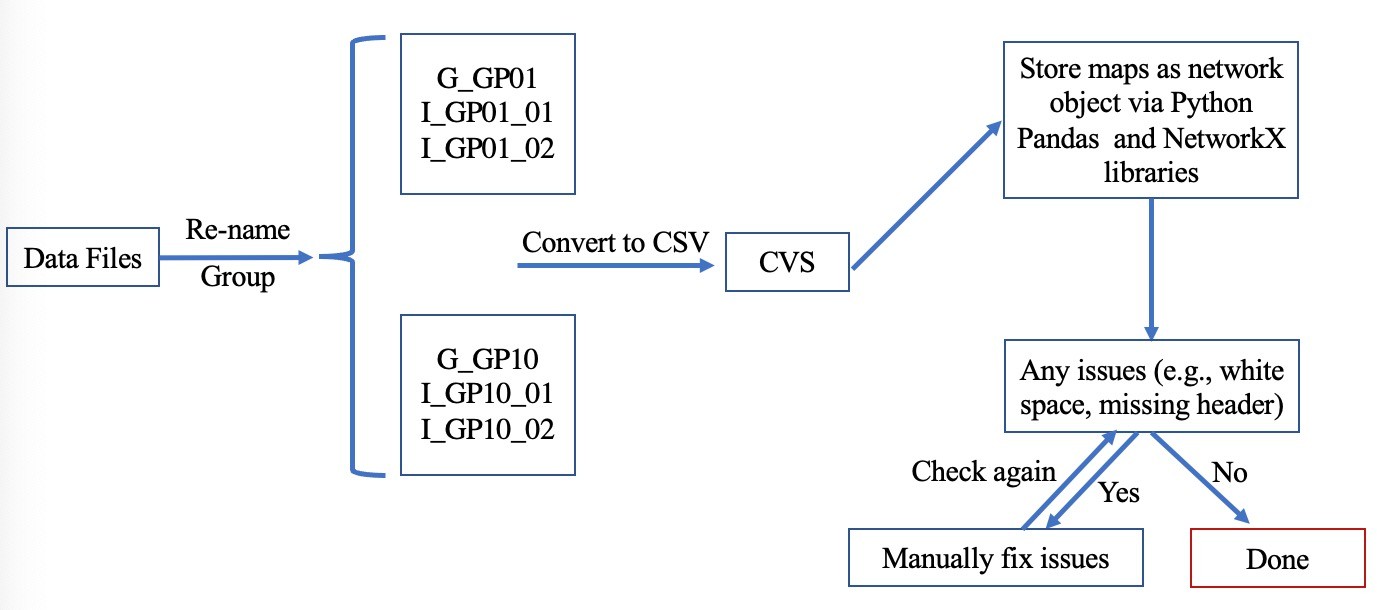


Figure 4.2: An overview of data cleaning process.

### Data Analysis

In this section, we applied network science to examine changes in various structural metrics that have been commonly applied when analyzing the structure of learners’ maps. This structure consists of a graph, which is defined as a pair *G* = (*V, E*), where *V* is the set of nodes or “concepts” and *E* is the set of edges or “relationships” [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [10](#_bookmark143), [11](#_bookmark144)].

The structure of a learner’s map can be analyzed at two levels, thus serving different purposes. *Graph-level metrics* emphasize the graph structure that represent the problem space [[5](#_bookmark138)] or the complexity of the problem [[87](#_bookmark220)]. Higher values of metrics, such as number of nodes, the number of edges, and the length of a map’s diameter, denote a larger scope of the problem space; while more cycles and the chains account for the complexity of the prob- lem. For *node-level metrics*, such as node centralities (e.g., eccentricity centrality, closeness centrality), they usually denote the importance of the concept and how it relates to other nodes [[4](#_bookmark137)]. However, these node-level metrics primarily serve the compare the focal points of “mental models” expressed as maps [[11](#_bookmark144)] rather than for the evaluation of learners’ knowledge understanding and map quality. Our analysis thus focuses on changes in graph-level metrics. Empirical studies have used many graph-level measurements [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [11](#_bookmark144), [46](#_bookmark179)]. We reflect this diversity and the complementarity of metrics by using 12 metrics in the data analysis (Table [4.1](#_bookmark82)). Figure [4.3](#_bookmark83) exemplifies the computation of graph metrics on a sample map from our data set. By applying the functions listed in Table [4.1](#_bookmark82) to each map, we obtain a table of measurements, of which a sample is shown in Table [4.2](#_bookmark84). Each line corresponds to one map, and columns track graph features of the map (e.g., number of nodes) as well as metadata on the setting leading to the map (e.g., whether it came from an individual G or a group I,

what was the group’s name and size).

To answer our first research question regarding which aspects of maps were impacted by a transition from individual to group mapping, we used one-way ANOVA test to identify metrics (in Table [4.1](#_bookmark82)) that show statistically significant differences (p-value<0.05) between individuals and groups. Next, to further explore how group maps were generated from the individual maps and how the identified aspects of a map structure were impacted, we analyzed the similarity of the maps by calculating the number of common nodes between group map and individual maps for each group. In this study, only identical terms are counted as common nodes. An example of similarity analysis is illustrated in Figure [4.4](#_bookmark85). Finally, based on the results from the one-way ANOVA test, Tukey’s HSD Test for multiple comparisons was conducted to answer our second research question: whether there is an effect of group size onto the metrics identified in the one-way ANOVA test. In the Tukey’s HSD Test, identified metrics serve as independent variables and the group size (individuals, 4 students, 5 students) as dependent variables.

Our methods were implemented in Python using a Jupyter Notebook. The notebook is publicly accessible on the Open Science Framework without registration needed at [https:](https://osf.io/rneku/)

[//osf.io/rneku/](https://osf.io/rneku/) in the folder named “TKNL (2022)”.

**Metrics Measures Implementation**

Number of nodes

Counted the number of nodes V(G) in a graph G.

(a.k.a Order)

Number of edges

Counted the number of edges E(G) in a graph G.

NetworkX’s

(a.k.a Size) Avg. degree Average number of edges per node |*E*(*G*)| .

|*V* (*G*)|

Density Number of edges existing as a ratio of

maximum edges 2|*E*(*G*)| .

|*V* (*G*)|∗(|*V* (*G*)|−1)

If a graph is connected, there is a path from

number\_of\_nodes(), number\_of\_edges() function

NetworkX’s

Connected

Diameter Number of

cycles Avg. length of cycles

Number of

paths Avg. path length (a.k.a Compactness)

Diameter of spanning tree

Number of components (a.k.a Ruggedness)

any node to any other node in a graph.

The furthest length between two vertices of the graph.

A cycle is a path that travels more than one node and ends at the starting point. Metrics include number of cycles and average length of cycles.

A path is the way that one node travels to another node. Metrics include number of paths and average length of paths.

Spanning tree is a portion of the graph where all nodes are visited once and the visiting path

is acyclic. The diameter is the shortest path between the two most distant nodes.

Components are disconnected parts of a graph.

is\_connected(G) function NetworkX’s shortest\_simple\_ paths(G) function

NetworkX’s cycle\_basis(G) function

NetworkX’s shortest\_simple\_ paths(G) function

NetworkX’s diameter(minimum spanning tree(G)) function NetworkX’s

number\_connected\_ components(G) function

Table 4.1: Descriptions of each graph metric, what the metric measures, and how they are implemented in this chapter.

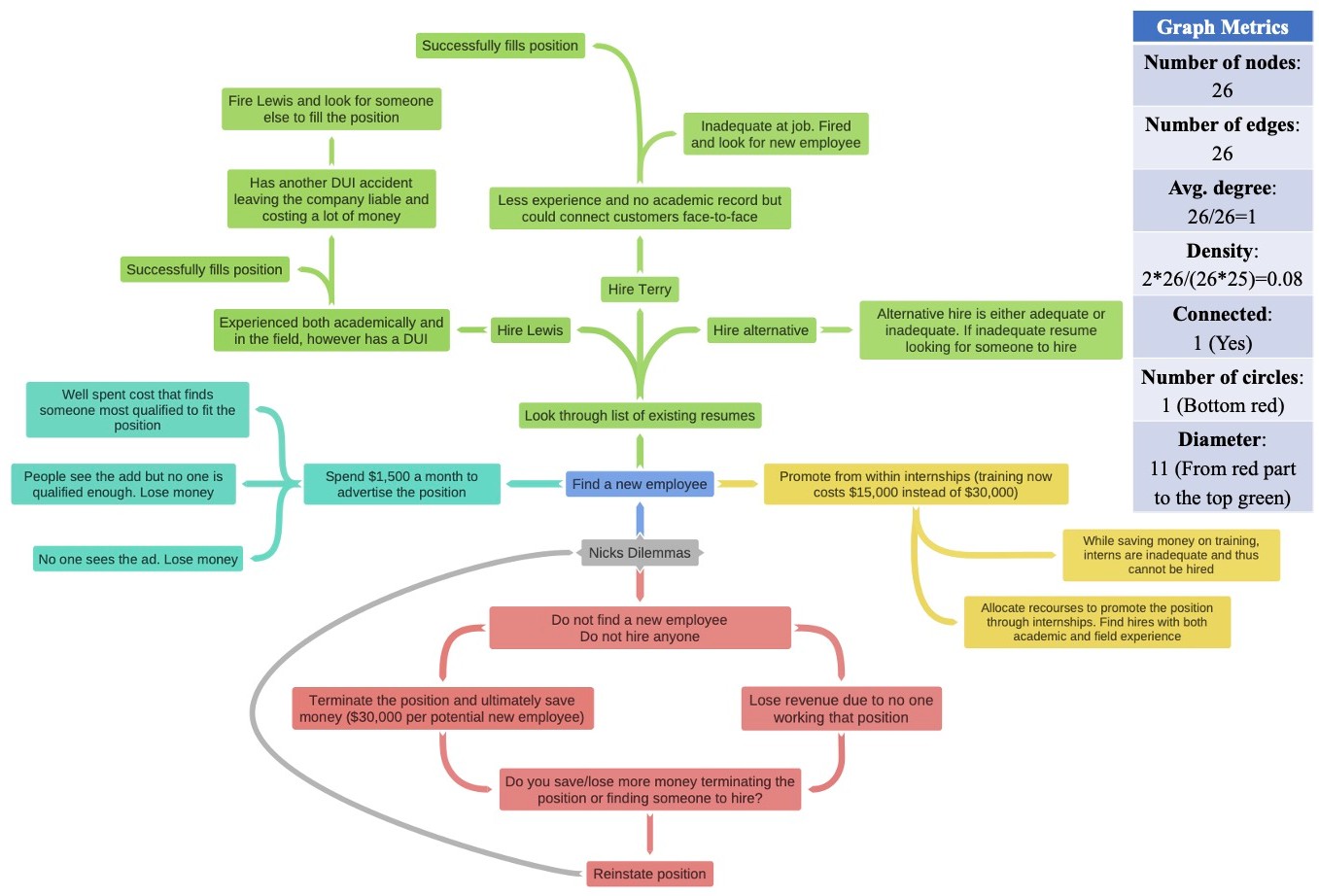


Figure 4.3: An example of how a map is evaluated via graph metrics and how the numeric values are produced. The map and its colors are produced by the software *Goggle*.

type

number of nodes

Maps

⋯

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| G\_GP04 | 30.00 | 33.00 | ⋯ | 3.19 | GP04 | G | 4.0 |
| I\_G\_GP04\_01 | 32.00 | 31.00 | ⋯ | 4.80 | GP04 | I | 4.0 |
| I\_G\_GP04\_02 | 21.00 | 20.00 | ⋯ | 3.72 | GP04 | I | 4.0 |
| I\_G\_GP04\_03 | 20.00 | 21.00 | ⋯ | 2.47 | GP04 | I | 4.0 |
| I\_G\_GP04\_04 | 14.00 | 13.00 | ⋯ | 2.79 | GP04 | I | 4.0 |

number of edges

avg. path length

group group

group size

Table 4.2: A sample of how maps and their graph metrics are documented in the spreadsheet.

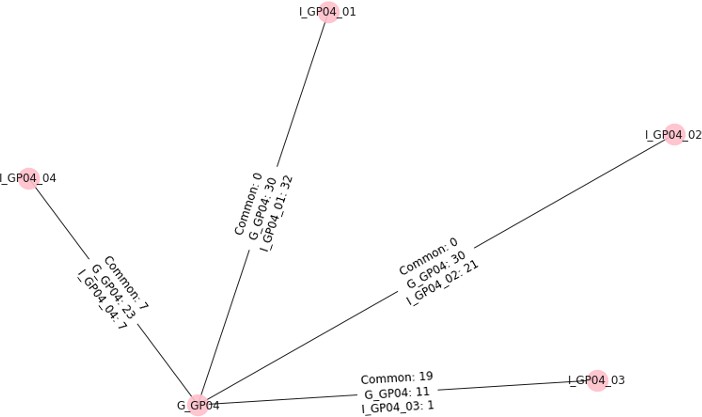


Figure 4.4: Similarity between collaborative map of group 4 and the maps previously pro- duced by its individuals. We perform three comparisons: “Common” counts the number of nodes shared between two maps, while the other two metrics denote the number of unique nodes (either unique to the group map G or to the individual map I).

## Results

### Q1: Transitioning from Individuals to Group Maps

Table [4.3](#_bookmark88) contrasts the structural metrics between individuals and groups of different sizes (4 and 5 students), where a green + signifies an increase of mean value from individual work to group work and a red – indicates a decrease. From this table, we noticed that the means of six graph metrics (number of nodes, number of edges, average degree, number of cycles, average length of cycles, and number of paths) increased from individual maps to group maps, whereas the mean *density* decreased.

The distributions for these metrics are shown in Figure [4.5](#_bookmark89), leading to the observation that two metrics (connected and number of components) do not change between individuals and groups since *all collected maps are connected* and hence have only one component. Consequently, we excluded these two metrics before performing the one-way ANOVA. Based on the results in Table [4.4](#_bookmark90), there were only five out of ten metrics with statistically significant difference (i.e., p<0.05) in mean value. These five metrics are: *number of nodes* (p=0.008),

95% Confidence Interval

Metric Size N Mean Std.

Deviation

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 44 | 25.57 | 11.53 | 1.74 | 22.06 | 29.07 | 11.00 | 58.00 |
| Number 4 | 6 | 40.50+ | 18.71 | 7.64 | 20.87 | 60.13 | 21.00 | 65.00 |
| of nodes 5 | 4 | 42.00+ | 25.05 | 12.52 | 2.15 | 81.85 | 16.00 | 76.00 |
| Total | 54 | 28.44 | 14.61 | 1.99 | 24.46 | 32.43 | 11.00 | 76.00 |
| 1 | 44 | 25.32 | 12.08 | 1.82 | 21.65 | 28.99 | 10.00 | 58.00 |
| Number 4 | 6 | 44.83+ | 18.26 | 7.45 | 25.67 | 63.99 | 24.00 | 69.00 |
| of edges 5 | 4 | 44.75+ | 30.29 | 15.15 | -3.45 | 92.95 | 15.00 | 87.00 |
| Total | 54 | 28.93 | 16.13 | 2.19 | 24.52 | 33.33 | 10.00 | 87.00 |

Std. Error

Lower Bound

Upper Bound

Min Max

Avg. degree

Density

Diameter

Number of cycles

Avg. length of cycles

Diameter of spanning

1 44 0.98 0.08 0.01 0.96 1.00 0.91 1.33

4 6 1.14+ 0.20 0.08 0.92 1.35 0.97 1.53

5 4 1.03+ 0.09 0.05 0.88 1.18 0.94 1.14

Total 54 1.00 0.11 0.01 0.97 1.03 0.91 1.53

1 44 0.10 0.04 0.01 0.08 0.11 0.04 0.19

4 6 0.07- 0.04 0.01 0.03 0.11 0.03 0.11

5 4 0.07- 0.04 0.02 0.00 0.13 0.03 0.13

Total 54 0.09 0.04 0.01 0.08 0.10 0.03 0.19

1 44 7.23 2.87 0.43 6.36 8.10 3.00 16.00

4 6 9.00+ 3.58 1.46 5.25 12.75 6.00 14.00

5 4 6.00- 1.63 0.82 3.40 8.60 4.00 8.00

Total 54 7.33 2.91 0.40 6.54 8.13 3.00 16.00

1 44 0.75 1.71 0.26 0.23 1.27 0.00 7.00

4 6 5.33+ 5.99 2.45 -0.95 11.62 0.00 17.00

5 4 3.75+ 5.68 2.84 -5.29 12.79 0.00 12.00

Total 54 1.48 3.18 0.43 0.61 2.35 0.00 17.00

1 44 1.31 2.68 0.40 0.49 2.12 0.00 12.00

4 6 3.33+ 1.67 0.68 1.58 5.09 0.00 4.65

5 4 2.02+ 2.33 1.17 -1.69 5.73 0.00 4.08

Total 54 1.58 2.61 0.36 0.87 2.30 0.00 12.00

1 44 7.39 3.11 0.47 6.44 8.33 3.00 18.00

4 6 9.67+ 3.27 1.33 6.24 13.09 6.00 14.00

5 4 6.25- 1.71 0.85 3.53 8.97 4.00 8.00

tree Total 54 7.56 3.11 0.42 6.71 8.41 3.00 18.00

1 44 379.00 372.50 56.16 265.75 492.25 55.00 1653.00

Number of paths

4 6 945.67+ 838.11 342.16 66.12 1825.21 210.00 2080.00

5 4 1096.25+ 1205.03 602.51 -821.22 3013.72 120.00 2850.00

Total 54 495.09 567.88 77.28 340.09 650.09 55.00 2850.00

1 44 3.80 1.02 0.15 3.49 4.11 2.07 6.30

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Avg. path 4 | 6 | 4.41+ | 1.42 | 0.58 | 2.92 | 5.90 | 3.03 | 6.32 |
| length 5 | 4 | 3.45- | 0.75 | 0.37 | 2.26 | 4.64 | 2.88 | 4.49 |
| Total | 54 | 3.84 | 1.06 | 0.14 | 3.55 | 4.13 | 2.07 | 6.32 |

Table 4.3: Descriptive statistics of graph metrics in maps.

*number of edges* (p=0.002), *average degree* (p=0.002), *number of cycles* (p<0.001), and

*number of paths* (p=0.005).

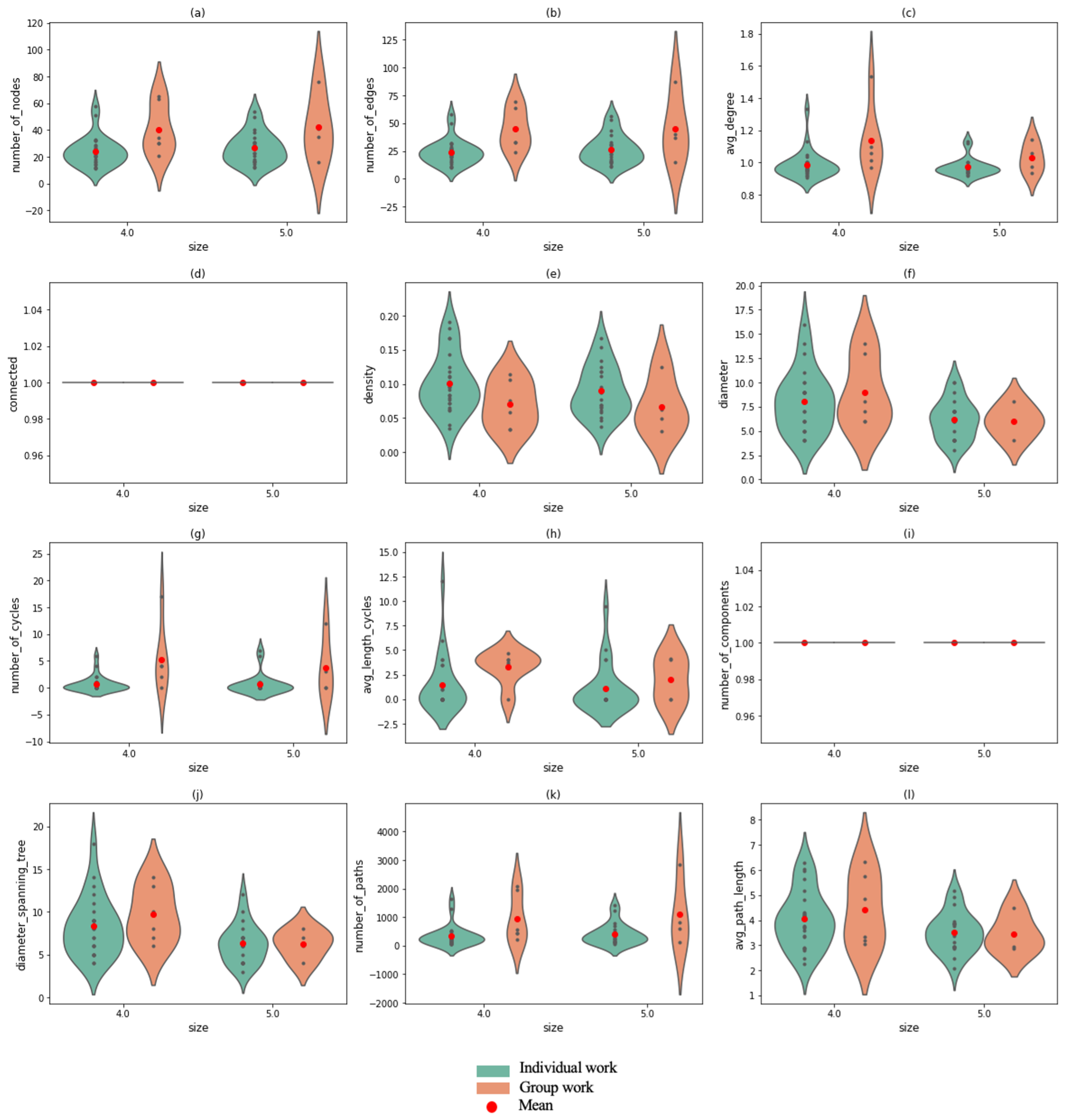


Figure 4.5: Distribution of graph metric values in both individual work and group work as well as the shift of mean values.

To understand how aspects of a map structure are impacted by transitioning from in- dividuals to groups, we examined the provenance of the concept nodes in the group map vis-à-vis the maps produced by individuals before group work. Given our examination of the provenance (Figure [4.6](#_bookmark91)), we see a clear divide between groups that *generate knowledge* compared to those that *integrate* it. First, there are groups with clear *knowledge generation*: the group-level map for five of the ten groups (Group 1, 3, 6, 8, and 9) was vastly different

#### Sum of Squares

**df**

**Mean Square**

#### F Sig.

Number of nodes

Number of edges

Between Groups 1971.04 2.00 985.52 5.38 **0.008\***

Within Groups 9344.30 51.00 183.22

Total 11315.33 53.00

Between Groups 3092.58 2.00 1546.29 7.38 **0.002\***

Within Groups 10689.13 51.00 209.59

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Total | 13781.70 | 53.00 | | | |
| Between Groups | 0.13 | 2.00 | 0.07 | 7.06 | **0.002\*** |
| Avg. degree Within Groups  Total | 0.48  0.61 | 51.00 | 0.01 |  |  |
| Between Groups | 0.01 | 2.00 | 0.00 | 1.92 | 0.157 |
| Density Within Groups Total | 0.08  0.09 | 51.00 | 0.00 |  |  |
| Between Groups | 24.27 | 2.00 | 12.14 | 1.45 | 0.243 |
| Diameter Within Groups  Total | 425.73  450.00 | 51.00 | 8.35 |  |  |

53.00

53.00

Number of cycles

Avg. length of cycles

Diameter of spanning tree

Number of paths

Avg. path

53.00

Between Groups 133.15 2.00 66.57 8.44 **<.001\***

Within Groups 402.33 51.00 7.89

Total 535.48 53.00

Between Groups 22.53 2.00 11.27 1.69 0.194

Within Groups 339.75 51.00 6.66

Total 362.29 53.00

Between Groups 34.82 2.00 17.41 1.86 0.167

Within Groups 478.52 51.00 9.38

Total 513.33 53.00

Between Groups 3256672.45 2.00 1628336.23 6.00 **0.005\***

Within Groups 13835058.08 51.00 271275.65

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Total | 17091730.54 | 53.00 | | | |
|  | 2.64 | 2.00 | 1.32 | 1.19 | 0.314 |
| length Within Groups | 56.71 | 51.00 | 1.11 |  |  |
| Total | 59.35 | 53.00 |  |  |  |

Between Groups

*\* The mean difference is significant at the 0.05 level.*

Table 4.4: Results of the one-way ANOVA test. Statistically significant differences (p <0.05) are bolded

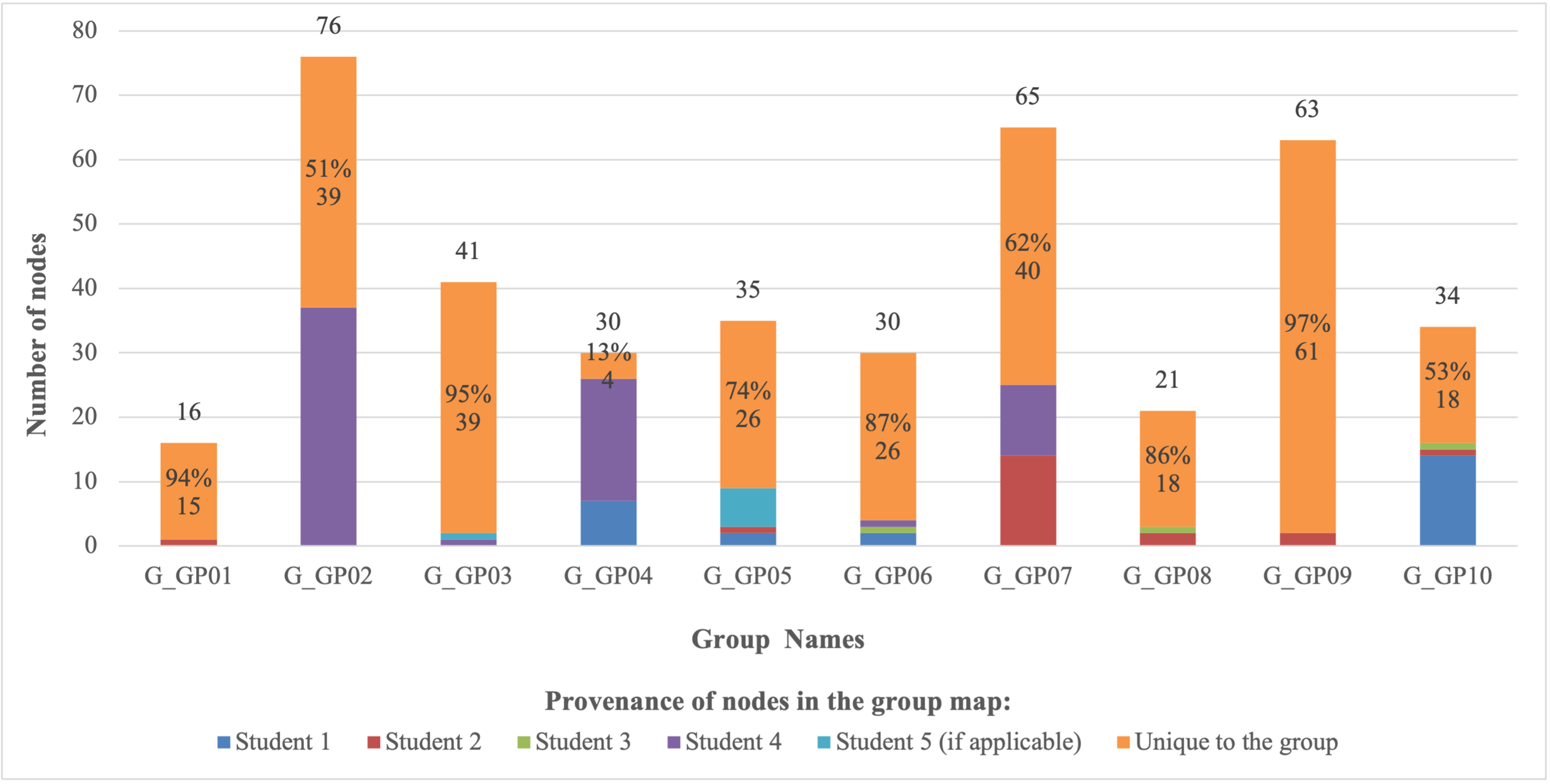


Figure 4.6: Similarity analysis of the composition for each group map. The number on top of each bar denotes the total number of nodes in the group map, the number in the yellow bar denotes the number of unique nodes distinct from any nodes in individual maps within the same group, and the percentage value in the yellow bar means the proportion of unique nodes in the group map.

from any individual map, with less than 15% of the group’s nodes originating from an indi- vidual’s map. Said otherwise, these groups demonstrate the generation of new knowledge as a group since 85% of their nodes appeared during the group mapping process. Second, some groups are more geared towards *knowledge integration*, with large shares of the group’s level map being attributable to specific individuals. The other five groups (Group 2, 5, 7, and 10) reused around 25%-50% nodes from individuals, typically one or two members in particular. For example, nearly half of the nodes in Group 2 came from the map of one student, while nodes in group 7 are mostly attributable to two students. Similarly, group 4 was nearly a combination of the map from two members, who produced 87% of the nodes.

### Q2: Potential mediating effect of the group size

We now shift our attention to our second line of inquiry (Q2): is the effect mediated by the size of a group? Table [4.3](#_bookmark88) shows that three metrics (diameter, diameter of spanning tree, and average path length) did not have clear trends when transitioning from individuals to groups, but were higher in the smaller group size (4) than in the larger groups (5). For the five metrics which showed statistically significant difference between individual and group maps, we conducted the Tukey’s HSD Test for multiple comparisons. Results in Table [4.5](#_bookmark93) show that the *number of edges* and *number of paths* have statistically significant differences

Dependent Variable (I) size (J) size Mean Difference

(I-J)

4

Std. Error Sig.

Number of nodes

Number of edges

Avg. degree

Number of cycles

Number of paths

1

Tukey HSD

5

1

Tukey HSD

5

1

Tukey HSD

5

1

Tukey HSD

5

1

Tukey HSD

5

4 **-14.932\*** 5.891 0.038

5 -16.432 7.069 0.061

1 14.932\* 5.891 0.038

5 -1.500 8.737 0.984

1 16.432 7.069 0.061

4 1.500 8.737 0.984

4 **-19.515\*** 6.300 0.009

5 **-19.432\*** 7.560 0.034

1 19.515\* 6.300 0.009

4

5 0.083 9.345 1.000

1 19.432\* 7.560 0.034

4 -0.083 9.345 1.000

4 **-.156\*** 0.042 0.001

5 -0.047 0.050 0.618

1 .156\* 0.042 0.001

4

5 0.109 0.062 0.199

1 0.047 0.050 0.618

4 -0.109 0.062 0.199

4 **-4.583\*** 1.222 0.001

5 -3.000 1.467 0.112

1 4.583\* 1.222 0.001

4

5 1.583 1.813 0.659

1 3.000 1.467 0.112

4 -1.583 1.813 0.659

4 **-566.667\*** 226.667 0.041

5 **-717.250\*** 272.000 0.029

1 566.667\* 226.667 0.041

5 -150.583 336.202 0.896

1 717.250\* 272.000 0.029

4 150.583 336.202 0.896

4

\* The mean difference is significant at the 0.05 level.

Table 4.5: Results of Tukey’s HSD Test.

between groups individuals, groups of four and groups of five students. For other metrics (number of nodes, average degree, and number of cycles), a statistically significant difference was only visible between individuals and groups of four students; there was no different between groups of four compared to five students.

## Discussion

Knowledge map, an effective tool to elicit and externalize learners’ knowledge struc- tures as graphs, is widely applied in the problem-based learning (PBL) setting to help learners solve complex problems (e.g., decision-making problems) individually and collab- oratively [[5](#_bookmark138), [55](#_bookmark188), [127](#_bookmark260)]. Although these studies have provided valuable insights on how map- ping techniques influence learning outcomes, there was a research gap on documenting the transformation of maps from individuals to groups. To examine this transition, we measured twelve characteristics of the causal maps produced by learners who first worked individually and then in groups of four or five students.

Our ANOVA (Table [4.4](#_bookmark90)) showed that only five out of twelve characteristics of the main have statistically significant changes, when transitioning from individuals to groups. These metrics generally denote the breadth of problem space (*number of nodes, number of edges, average degree*) and the structural complexity of knowledge conceptualizations (*number of cycles, and number of paths*) [[137](#_bookmark270)]. The wider scope of the problem space not only indicates that participants are able to identify more important elements or factors to consider in the decision making process, but also provide explanatory linkage or relationships to articulate the reasons why the elements or factors are essential to attain the solution to the prob- lem [[124](#_bookmark257), [127](#_bookmark260)]. In addition, the increase of *number of cycles* and *number of paths* means that participants could examine the problem in a larger context and explore more paths to the solution [[5](#_bookmark138), [87](#_bookmark220)]. This finding provides concrete characterizations of the more sophisticated maps that students are able to achieve as groups compared to individuals, hence quantifying the benefits of collaborative mapping for performances in problem-based learning.

An important part of problem-based learning is that it involves learner-learner interac- tions, hence groups define the problem, analyze factors, generate hypotheses, and identify deficiency [[114](#_bookmark247), [138](#_bookmark271)], as shown in learning cycle (Figure [4.1](#_bookmark75)). Studies suggest that the group size may influence the dynamic of learner-learner interaction and thus affect the learning performance [[139](#_bookmark272), [140](#_bookmark273), [141](#_bookmark274)]. Our work-in-progress study provided some evidence on the impact of group size on map construction. The results of Tukey’s HSD Test (Table [4.5](#_bookmark93)) in- dicated a *tendency* that more graph metrics presented significant changes in smaller groups in the transition from individual to group mapping. However, this aspect was limited by the small variability in group sizes (four and five students), hence our methods would have to be applied to a large and more varied sample in the future.

Our similarity analysis between the map of a group and the previous maps of its in- dividual members showed two broad strategies. Half of the groups engaged in knowledge creation, with the group-level map containing a majority of unique concepts that could not be attributed to any particular member. The other half of the groups adopted a knowledge

integration approach, as the maps of a few members (often one or two) clearly resulted in the product of the group. As Freund and Giabbanelli [[88](#_bookmark221)] discussed, teams that aggregate individual knowledge into a shared representation may not systematically report how their members understand and interpret their original sources. Hence this study only observes the dichotomy, but we cannot yet explain why groups picked different strategies.

There are multiple ways to build upon this work-in-progress study for future exploration. First, our sample size was limited by the enrollment of students in a marketing course, hence we have relatively small and imbalanced sizes of groups (6 groups with 4 members and 4 groups with 5 members) for the comparative analysis in one-way ANOVA test. As our methods have demonstrated the feasibility of tracking changes in maps between individuals and groups, future studies can increase the sample size to have more students and also a greater variability in group sizes. We also note that our study was performed on a decision- making problem pertaining to marketing, hence future studies may assess whether there is a disciplinary effect.

Second, our study only contained data on maps of individuals and maps of the groups. Students were able to start the conversation on the discussion board and many opted to continue it onto Zoom, hence we do not have complete recordings of all the actions leading to the group map. The process of transitioning from individual views to group-level models has been documented in previous studies [[142](#_bookmark275)], hence future work may ask students to operate solely within a controlled environment to examine the sequence of operations leading to the group map. The exploration of the transition process may provide a clearer explanation on how and why students adopt different strategies in constructing group maps.

Last, our questions were about changes in the *structure* rather than the *content* of the maps. Changes in content may have happened, for example by improving the grammar, addressing spelling errors, or transitioning to a professional vocabulary that is more appro- priate in the application domain [[143](#_bookmark276)]. Future research may examine the terminology used by students, for example by tracking linguistic variability to assess whether the same notion is now portrayed in a more professional manner by the group. A particularly interesting possibility is to jointly examine changes in structure and content, as they can affect each other when transitioning from individuals to groups.

## Conclusion

For the study conducted in this chapter, we analyzed 54 knowledge maps, including 44 individual maps and 10 collaborative maps from 10 groups, and documented the changes of 12 structural metrics (e.g., number of nodes and edges, diameter) in the transition from individual to group mapping. Results suggest that graph metrics generally denoting the breadth of problem space (*number of nodes, number of edges, average degree*) and structural complexity of knowledge conceptualizations (*number of cycles, and number of paths*) present statistically significant difference between individual and group work. However, similar group sizes do not have significant differences on those changes of graph metrics for our cases, hence the sample of this work-in-progress study does not recommend between forming groups of

four or five students.

**Chapter 5**

# Algorithmic Approaches to Guide a Student’s Map Using Multiple Expert Maps

## Introduction

Knowledge maps have been widely used for supporting students’ meaningful learning and served as an effective tool for formative assessment [[3](#_bookmark136), [5](#_bookmark138), [41](#_bookmark174), [45](#_bookmark178)]. Through knowledge maps, students are able to elicit and externalize their organization of knowledge structures and in- structors could examine the externalization of knowledge maps to provide useful information for guiding students’ learning [[3](#_bookmark136), [4](#_bookmark137)].

One common way of assessing knowledge maps is the *reference-based* approach, which compares a student’s map with an expert’s map and scores the differences and similarities between them [[4](#_bookmark137), [7](#_bookmark140), [52](#_bookmark185)]. Currently, multiple software offer techniques to assess a map by comparing it with an expert’s work (Table [5.1](#_bookmark98)). Despite the success of these software in capturing the differences between maps [[3](#_bookmark136), [144](#_bookmark277)], applying graph-based assessment for sum- mary writing [[8](#_bookmark141), [148](#_bookmark281)], and providing automatic feedback in map construction [[37](#_bookmark170), [145](#_bookmark278)], a shortcoming of the reference-based evaluation is the constrain of comparing the student’s map with *only one single expert map* at a time. This implicitly provides students with the assumption that there is only one standard answer to the question and only one way to think correctly. However, in reality, people’s thinking is always diverse, complex, and con- textualized, especially for ill-structured problems and open-ended questions [[13](#_bookmark146), [14](#_bookmark147)]. Studies about acquiring and assessing knowledge from multiple experts also show that experts have different ways of understanding a problem and expertise conflicts between the experts is an inevitable phenomenon [[151](#_bookmark284), [152](#_bookmark285)]. Problems related to decision making, policy making, and debate questions are common examples that admit multiple perspectives and answers. It is uncommon to see that there is a single correct answer, solution, or viewpoint to those questions. The nuances in thinking would thus result in different knowledge map structures. For example, consider the following open-ended question: should guns be banned in the United States? The knowledge maps constructed by people who advocate for “yes” would have differences with maps of experts arguing for “no”, while other experts may include nu- anced viewpoints with certain conditions and hence even greater differences in their maps’ content. If students’ knowledge maps are evaluated and their learning is promoted based on one single expert map, students would be constrained to only that viewpoint and hence

**Software Comparison Methods References**

HIMATT

GIKS

ITACM

SMART

SMD

jMAP

M-tool

HIMATT analyzes and assesses a student’s map mainly through calculating and comparing the descriptive measures for knowledge representations with an expert map. Measures include six structural indicators: connectedness, ruggedness, average degree of

vertices, number of cycles, nodes, and edges.

GIKS captures semantic knowledge structures from students’ writing and converts them into network graphs. By comparing the

student’s map with an expert map, the identical, missing, and incorrect links of concepts in writing are identified and marked. The comparison with identified similarities/differences thus provides information of students’ knowledge strengths and weaknesses and guidance for learning improvement.

ITACM compares a student’s with an expert’s maps through various approaches (e.g. graph kernels) to measure the similarity of maps.

Via the structural analyses of both student and expert maps, it automatically generates feedback to guide the student’s modification of the map.

SMART integrates a new index of Graph Centrality with other indices derived from the 3S model (surface, structure, semantic) [[146](#_bookmark279), [147](#_bookmark280)] in the comparison analysis between a student map and an expert map.

Individual feedback, including an expert’s reference summary and a visualized comparison will be delivered to the student that prompts modification towards a more cohesive and solid mental representation similar to the expert.

SMD is designed based on the theory of mental model and graph theory with the purpose of measuring relational, structural, and semantic

levels of graphical representations and concept maps. It assesses knowledge maps through surface components of graph (reference-free), structural (reference-free), and semantic matching (reference-based). jMAP provides functions that mine data from students’ causal mapping behaviors to compare and identify the differences in terms of action sequences for map construction.

M-tool is a standardised and inclusive mental model mapping tool

that supports participants to create graph diagram with nodes representing system variables and weighted arrows displaying nodes’ relationship.

By identifying differences and/or changes in mental models, M-tool helps stakeholders to develop strategies to tackle challenges within the system.

Table 5.1: A list of software for map comparison.

[[144](#_bookmark277)]

[[55](#_bookmark188), [8](#_bookmark141)]

[[3](#_bookmark136), [145](#_bookmark278)]

[[148](#_bookmark281)]

[[7](#_bookmark140)]

[[149](#_bookmark282)]

[[150](#_bookmark283)]

would be unable to engage in holistic cognitive thinking.

Furthermore, research regarding human learning demonstrates that providing students with multiple sources of information (e.g., compatible and conflicting) as well as support and

challenge in the learning process is particularly beneficial in developing their critical think- ing and improving their knowledge understanding [[14](#_bookmark147)]. Empirical studies show that the iteration of support and struggle can assist students in navigating the challenges associated with problem-based learning and provide students with opportunities to provoke and explore richer conceptual understanding in learning [[153](#_bookmark286), [154](#_bookmark287)]. Giabbanelli and Tawfik’s study [[5](#_bookmark138)] indicates that students exposed to failure cases (contradictory to students’ initial thoughts) demonstrate broader scope in map construction compared to their peers exposed to success study cases (confirming students’ initial thoughts). Thus, it is important to consider and integrate multiple perspectives (similar or dissimilar) into knowledge maps assessment sys- tems to provide constructive feedback to students. One core issue of the map assessment system is how to handle the multiplicity of experts. In this chapter, we discuss five distinct methods to measure the distance or “difference” between maps; and based on the different distance measurement, we develop an algorithmic frameworks that include four comparison strategies to capture and use the differences to guide students’ learning.

The remainder of this chapter is organized as follows. In Section [5.2](#_bookmark99), we introduce various methods in current map comparison systems that use reference-based approaches to measure the difference between maps. Next, we present our design of algorithmic frameworks for capturing the similarities/differences between a student map and multiple expert maps in Section [5.3](#_bookmark107). Lastly, we discuss pros and cons of the above design and potential developments in future work.

## Reference-based Methods for Map Comparison

In general, there are two types of map comparison: exact match and inexact match. Exact match is derived from the *graph isomorphism* problem in graph theory. Two graphs *G*1(*V*1*, E*1) and *G*2(*V*2*, E*2) are isomorphic if there exists a one-to-one correspondence mapping *F* between the vertex sets of *G*1 and *G*2 such that any two vertices *u* and *v* of *G*1 are adjacent in *G*1 if and only *F* (*u*) and *F* (*v*) are adjacent in *G*2 [[155](#_bookmark288)]. However, in practice the student and expert maps are very unlikely to be identical. What educators or students care about in formative assessment is the extent to which their maps are similar, which thus prompt studies on inexact matching methods. One common approach to handle inexact matches is incorporating the concept of *matching cost* in the matching algorithm to penalize the structural differences existing between graphs [[145](#_bookmark278), [156](#_bookmark289)], where two graphs are considered more similar if the matching cost is lower. A frequently used method of applying matching cost is the graph edit distance (see Section [5.2.3](#_bookmark102)). In this section, we do not seek to cover all comparison methods exhaustively. Instead, we focus on common comparison methods which would be covered in our framework design.

### Global Graph Metrics

The overall similarity between two graphs can be easily computed by simply comparing the graphs’ global measurements, such as degree distributions, graph diameter, and average

path length (see more metrics in Table [3.1](#_bookmark53)). However, these graph metrics usually capture limited aspects of complex graphs, hence similar values of those metrics do not necessarily indicate similar structure of the graph [[157](#_bookmark290), [158](#_bookmark291)]. For example, two graphs with similar diameters may contain completely different sets of nodes and edges. Thus, the comparison of global graph metrics often provides a first analysis of two graphs.

### Adjacency Matrices

Another simple way to compare two graphs is directly computing the difference of the adjacency matrices of the two graphs. This straightforward method allows users to select the matrix norms based on the demand of the research domain. For example, one common norm used for the proximity measurement to compute the similarity between two objects is

Jaccard distance [[158](#_bookmark291), [159](#_bookmark292)]. Given two distinct graphs *G*1(*V*1*, E*1) and *G*2(*V*2*, E*2), we could represent the graphs as a *n* × *n* adjacency matrix *A* in terms of nodes, where *n* is the number of nodes in the graph. The element in the matrix [*aij*] epresents the weight of the edge from node *i* to *j*, which is 0 in the absence of such an edge. For *G*1 and *G*2 with adjacency matrices

*A*1 = [*a*1 ] and *A*2 = [*a*2 ] respectively, and identical node sets *V* = *V*1 = *V*2 (if *V*1 /= *V*2 ,

*ij ij*

we take *V* = *V*1 ∪ *V*2 and pad with zeros the adjacency matrices), the Jaccard Distance is defined as [[158](#_bookmark291), [159](#_bookmark292)]:

*dJ* (*G*1*, G*2) = 1 — *J*(*A*1*, A*2) = 1 — *A*1∩*A*2

*A*1∪*A*2

where *J*(*A*1*, A*2) is the Jaccard similarity/coeﬀicient and it is computed as:

( ∑*i,j*∈*V min*(*a*1 *,a*2 )

∑

*ij*

*ij*

1

2

*i,j*∈*V max*(*a*1 *,a*2 )

*i,j*∈*V*

*ij*

*ij*

*if* Σ

*max*(*a , a*

) *>* 0

1 2 1 *if* Σ

*J*(*A , A* ) =

*ij*

*ij*

*max*(*a*1 *, a*2 ) = 0

### Graph Edit Distance

*i,j*∈*V*

*ij ij*

*Graph edit distance (GED)* is one of the most employed methods to measure the similarity between two graphs in inexact graph matching [[145](#_bookmark278), [156](#_bookmark289), [160](#_bookmark293)]. Unlike exact matching, differences between graphs in inexact matching are allowed but would be penalized, thus it generates the matching cost. In the GED, a graph (student map) can be transformed to another one (expert map) within a finite sequence of graph edit operations. These operations may include node or edge insertion, deletion, and relabeling. Each operation would result in an edit cost defined in the cost function, and the sum of costs for all operations is the total matching cost. Although there are many possible sequences of edit operations to transform one graph to another graph (e.g., deleting the student map and copy the expert map from scratch), GED is concerned with finding the sequence of edits with least cost, which becomes the best edit path. In the example of Figure [5.1](#_bookmark103), the optimal transformation path would follow the green or orange sequence of operations if all operations are assumed to cost the same.

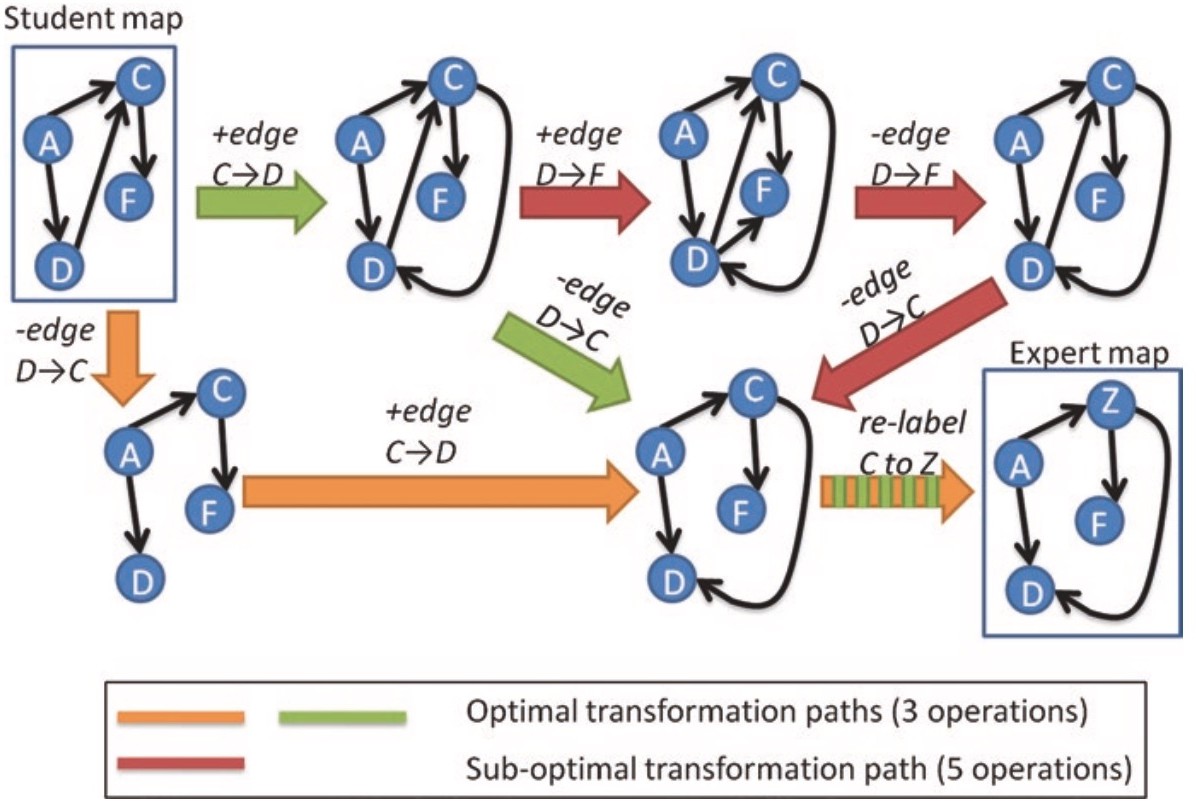


Figure 5.1: An example of GED. Reproduced from Giabbanelli et al [[145](#_bookmark278)].

### Graph Embedding

As mentioned in Section [5.2.2](#_bookmark101), various global graph metrics derived from graph theory could be used for graph comparison. However, the simple comparison of each individual metrics do not always provide robust results, because similar values in one graph metric do not always mean that the two graphs are similar. Thus, a more comprehensive score is needed to truly reflect the similarity of the graph structure. Such composite score can be calculated based on a set of graph selected metrics. The idea of graph embedding is to represent the graph in a vector space and the distance between the two embedded graphs represents the similarity of those two graphs. For instance, we could represent three graphs through three different graph metrics and position them as a point in a 3D space. The distances between points indicate their similarities (Figure [5.2](#_bookmark106)).

### Tversky Similarity

Although the structural graph metrics in the above methods produce important informa- tion regarding the knowledge organization for comparison, they lack the ability to account for the semantic meaning of the knowledge, which is a central part in explaining knowledge representation [[52](#_bookmark185)]. In many existing knowledge map assessment tools (e.g., SMD [[7](#_bookmark140)], HI- MATT [[144](#_bookmark277)], MITOCAR [[161](#_bookmark294)]), Tversky similarity [[53](#_bookmark186)] (see Section [2.3.3](#_bookmark33)) is implemented as a metric for quantifying semantic similarity between two graphs.

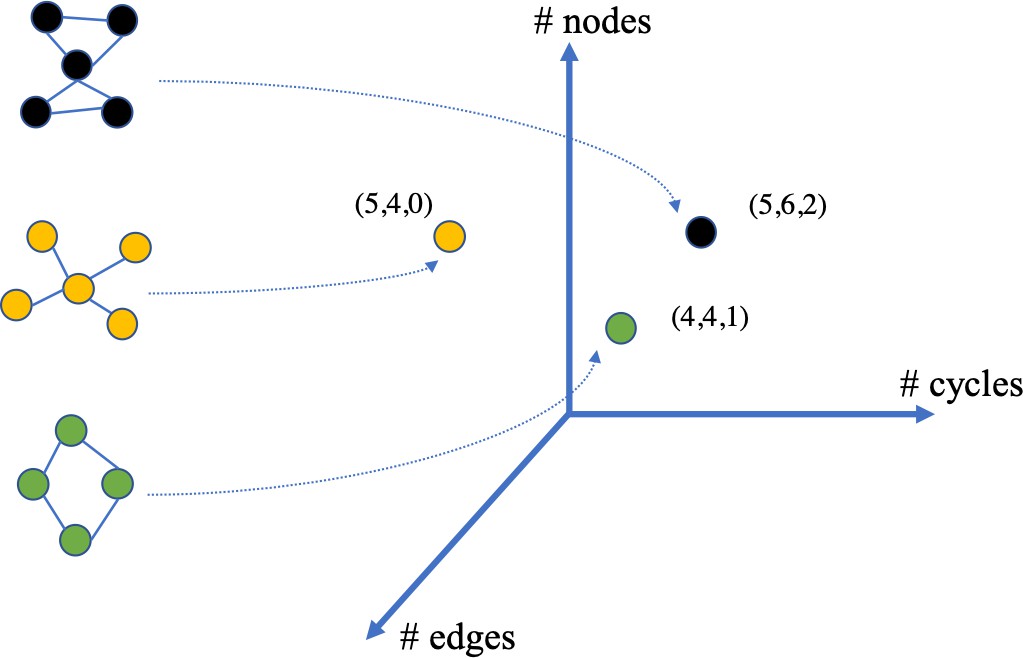


Figure 5.2: An example of graph embedding in 3D space by measuring three graph metrics: number of nodes, edges, and cycles.

## Framework

In this section, we will present our algorithmic framework on how to handle the multi- plicity of expert maps in the knowledge map assessment. Intuitively, the framework would grant students access to their own maps and multiple expert maps, provide them with the comparison results between maps as feedback, allow them to modify maps to bridge the knowledge gap with the guidance, and show them the effects after each modification. Stu- dents are able to iterate the process and take actions to respond to the feedback they receive from the system. Thus, the task of creating such a framework requires to answer three major questions: (a) which method of similarity calculation should be selected? (b) which expert maps should be presented to students? (c) what kind of feedback should be provided to students? The answers are not necessarily to be unified in our framework, rather, they are affected by the strategy (see Section [5.3.2](#_bookmark110)) adopted in the comparison process between the student’s map and expert maps.

### Main Elements in the Framework

The main elements of the framework are listed in Table [5.2](#_bookmark109) and each element will be ex- plained in this section. In the framework, a student’s knowledge structure is represented as a directed knowledge map *G*(*V, E*), where *V* denotes the knowledge concepts and *E* means the relationship among those concepts in student’s understanding. This graph representa-

|  |  |
| --- | --- |
| **Notation** | **Meaning** |
| *G*(*V, E*) | A student map, consisting of nodes *V* and directed links  *E*. |
| E | A list of expert maps in graph form. |
| P | A list of control parameters (e.g., number of iteration, set  of graph metrics, similarity distance function) |
| *S* | A comparison strategy used in map comparison. Given  the student map *G* and expert sources E, it specifies how expert maps will be involved in the process, parameters used, and how the student will respond to the feedback. |
| *O* | Outputs, final student map *G*∗, and the documentation  of actions *A* the student made. |

Table 5.2: Main elements of the framework.

tion helps students to articulate their understanding of the problem space and its possible relationships [[12](#_bookmark145), [145](#_bookmark278)]. The collection of expert maps available to the student in the as- sessment is denoted as E = {*e*1*, . . . , en*}. Note that the (sub)set of experts presented to a student depends on the choice of comparison strategies as well as the students’ actions, hence different maps may be shown as the execution of the algorithm unfolds. A list of control parameters that guide and constraint the student’s actions is expressed as P = {*p*1*, . . . , pn*}. These parameters include similarity distance function, selected graph metric(s), iteration of the process, ratio of similar and dissimilar expert maps exposed to the student, etc. Choices of the parameters are based on the problem characteristic and knowledge domain, especially

for the selection of graph metric(s) (see Chapter [3](#_bookmark49)). Given a student map *G*(*V, E*) and the list of the expert source maps E as input, the selected strategy *S* specifies the steps of comparing the student map with expert maps, control parameters used in the process, how the student may respond (i.e., potential actions) to the feedback from the comparison, and the final output. The strategy *S* can be denoted as a function:

*S* : (*G,* E*,* P)

'→ *O* ( *P* ⊆ P

*, E* ⊆ E )

a student map`, exp˛e¸rt mxaps, parameters

o`u˛tp¸uxt select`ed ˛pa¸raxmeters

selecte`d e˛x¸perxt maps

### Four Distinct Comparison Strategies

##### Strategy One: Towards Similar

The first approach is simple and straightforward as an extension of current techniques to compare maps (e.g. GIKS, HIMATT), which consider that two maps are more similar if they share many common connections [[7](#_bookmark140), [12](#_bookmark145)]. This approach takes in a student map, a list of expert maps, and three parameters (the number of similar expert maps to expose *k*, the iteration time *t*, and the similarity distance function *f* ). These parameters are set by the instructors based on facts such as knowledge domain and total number of expert maps.

Instead of using only one expert map in the assessment, we compute all similarity distance between the student map and each expert map, select *k* expert maps that are most close to the student map, expose them to the student, ask the student to take some of the differences into account to modify the map, and then re-compute which expert maps are closer as an iterative process. Finally, it will output the final revised map and the documentation of all edits made. Figure [5.3](#_bookmark111) illustrate the detailed process by way of an example.

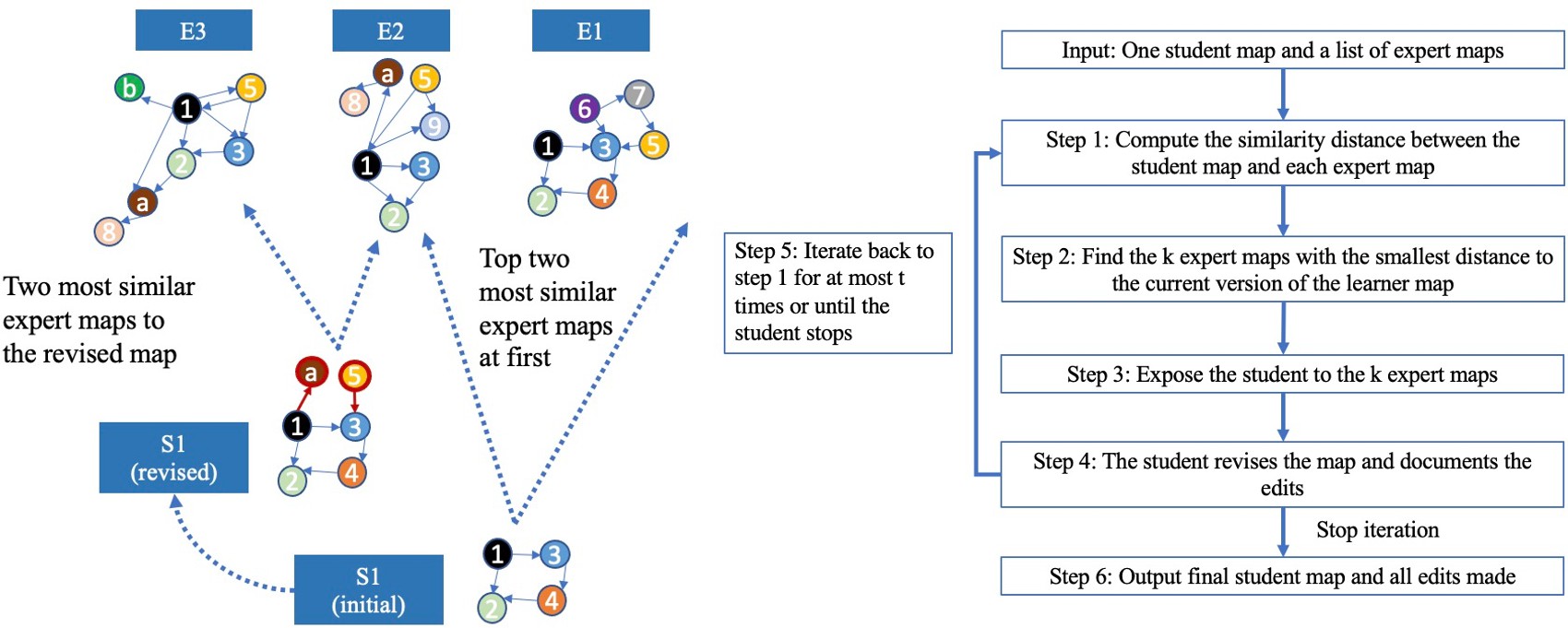


Figure 5.3: Detailed process and an example of *Strategy One*. Red links and circles in the revised student map (left figure) mean the new edits.

##### Strategy Two: Combine Similar and Dissimilar

*Strategy One* gives us a baseline direction for handling the multiplicity of expert maps, but showing only similar maps to the student may result in the same issue (see Section [5.1](#_bookmark97)) as comparing it with one single map. For instance, during the iterations, the maps exposed to the student may be only the most similar ones, hence they will exclusively serve to confirm the student’s initial ideas. Maps containing contradictory opinions that are able to promote deeper thoughts [[5](#_bookmark138), [144](#_bookmark277)] may be screened out due to the comparison mechanism. Motivated to improve this limitation, *Strategy Two* is created to incorporate both consonant and dissonant cases.

In the second strategy, the student is exposed to *n* most dissimilar expert maps first and asked to react to those differences from the expert maps. After the first round of modifications, the student is then exposed to *m* most similar expert maps and asked to react again. The detailed process along with an example is shown in Figure [5.4](#_bookmark112). Besides the parameters (*k*, *t*, and *f* ) explained in *Strategy One*, we add a new parameter *α* to control the ratio of dissonant/consonant cases to expose, namely *n*/*m*. The value should be set to *α <* 1 because we expect to use a few dissonant cases to prompt deeper thoughts about the system, but more consonant cases to avoid creating discomfort by accumulating only dissonant ones [[162](#_bookmark295), [163](#_bookmark296)].

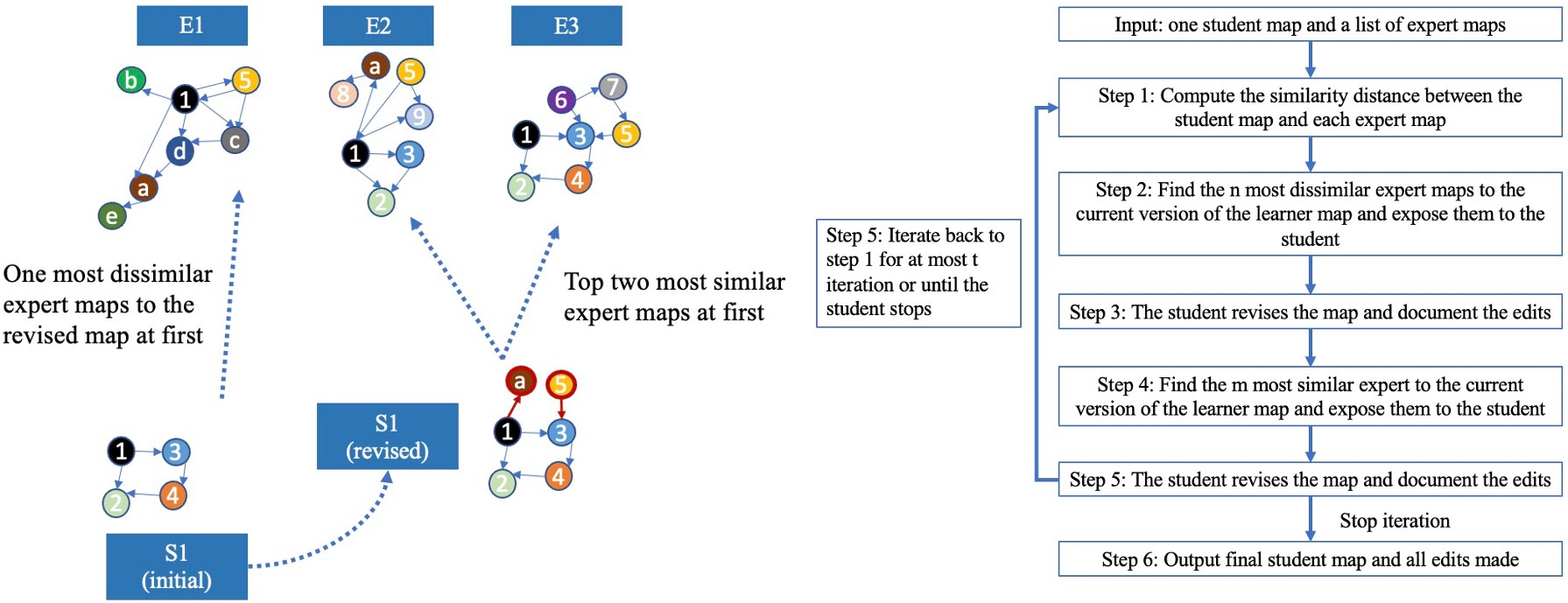


Figure 5.4: Detailed process and an example of *Strategy Two*. Red links and circles in the revised student map (left figure) mean the new edits.

##### Strategy Three: Action Sequences with GED

Jeong showed that students could benefit from predefined specific action sequences (e.g., move node→link node→redirect link) in constructing more accurate causal maps and achieve deeper causal understanding of the concepts [[149](#_bookmark282)]. In addition to the action sequence, imme- diate feedback in the assessment system would promote students’ learning and correct their first unsatisfactory response [[164](#_bookmark297), [165](#_bookmark298)]. These two observations are important to support students in assessing how closely their map resembles the expert maps, and to transform it in a reasonable manner.

Therefore, to integrate the notions of action sequences and immediate feedback, *Strategy Three* adopts a combination of GED with another similarity distance function (different from GED). In this approach, the student will be exposed to paths with action sequences gen- erated based on GED instead of just showing *k* similar/dissimilar expert maps. Figure [5.5](#_bookmark113) demonstrates how this strategy works along with an example. As step 2 and 3 show, the student is provided with a list of options, where the student can see the path to transform to each expert maps with total edits to make as well as how similarity would change after performing the first edit in each option (Figure [5.6](#_bookmark114)). Before the student makes any modi- fication, he is able to learn the expected effect (e.g., similarity increase or decrease) caused by the suggested edit he hopes to make, namely the first edit from GED. This will give the student a broader picture of how the revised map would transform, hence allowing the student to decide his personalized edit path. The process repeats until the student decides to stop. The parameters for this strategy include the iteration time (*t*) and the similarity distance function (*f* ) other than GED.

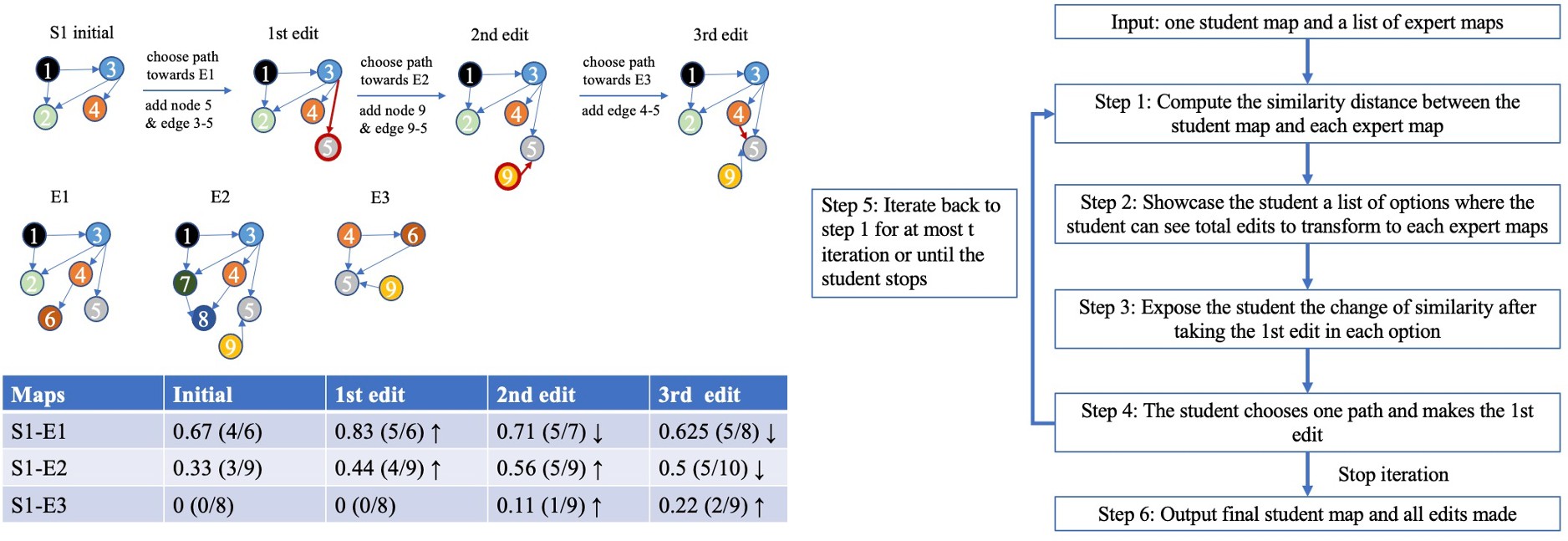


Figure 5.5: Detailed process and an example of *Strategy Three*. In this example, the similarity is calculated as the ratio of shared edges. Changes of similarity are provided to the students for each edit so that the student understand whether his edit would shrink or enlarge the gap to the experts’ maps.

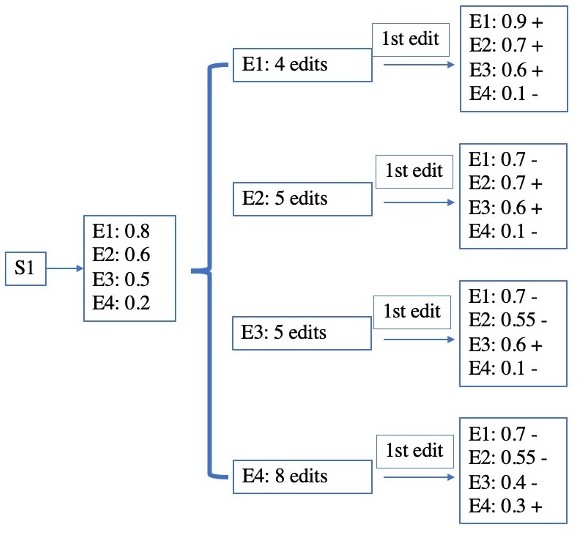


Figure 5.6: Different paths with information of total edits and the change of similarity to each expert map in each path, + means increase and — means decrease.

##### Strategy Four: Cluster-based Comparison

Similar to *Strategy Three*, *Strategy Four* seeks to provide the student with a broader view on how his map is shifting towards the expert maps. Instead of using GED as feedback information, we cluster the expert maps based on graph-level characteristics via graph em- bedding. The selection of graph-level features based on problem characteristic is studied in Chapter [3](#_bookmark49). Then, we expose the student to the summary statistics of those graph metrics for each group and ask the student to react to those differences by modifying his map. As changes are made, we show the student which of the groups he gets closer to. This process will iterate at most *t* times or until the student believes the edits he made are suﬀicient com- pared to the expert maps. Via the visualization of the trajectory, the student is able to know whether he is heading in the right direction, that is, towards certain groups of experts [[145](#_bookmark278)].

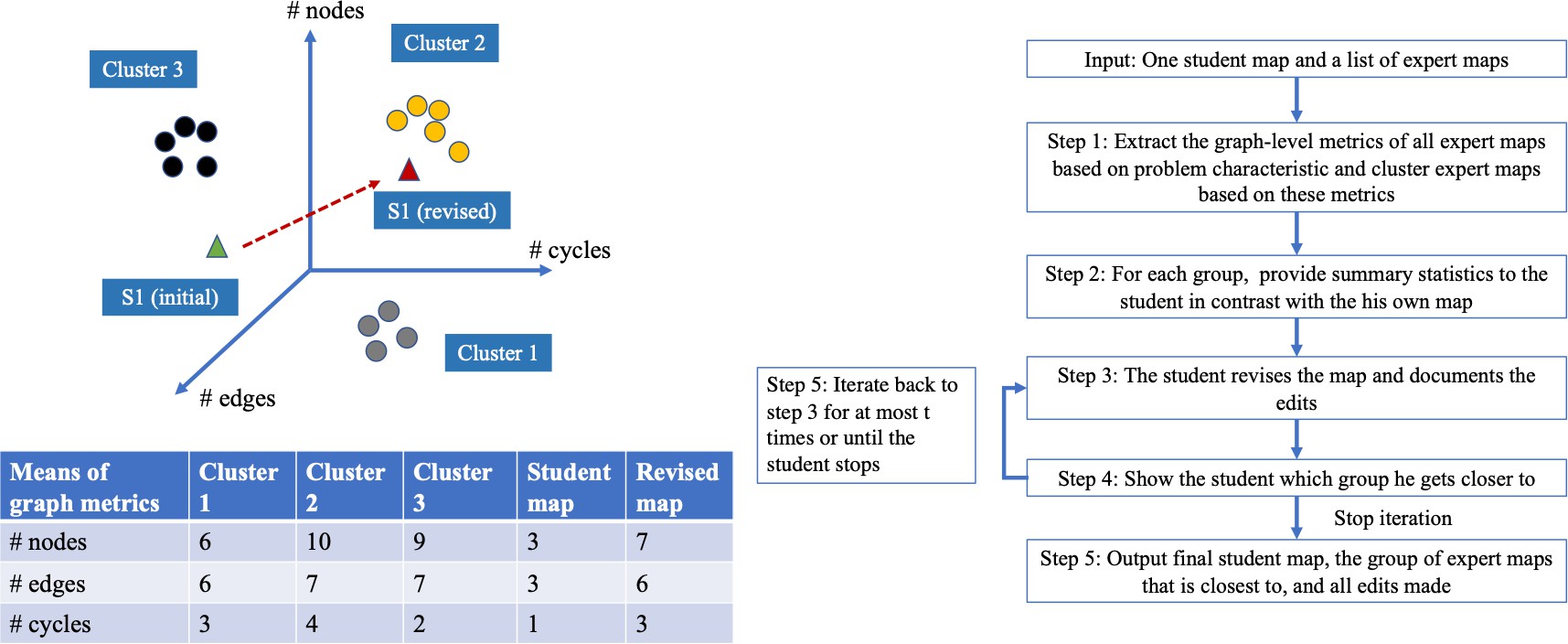


Figure 5.7: Detailed process and an example of *Strategy Four*. The effect of edits is presented as the moving track of the triangle (student map). As a result, the student map is moving towards the expert maps in cluster 2.

We illustrate this method by positioning maps in a 3D space via three graph metrics (number of nodes, number of edges, and number of cycles) (Figure [5.7](#_bookmark115)). In the example, we embed the 14 expert maps into a 3D space consisting of three axes representing the above three metrics respectively and cluster them into three different groups through clustering algorithms (see Chapter [3](#_bookmark49)). Then we expose the student to the mean statistics of selected graph metrics. Then we position the student map in the same space and ask the student to revise the map based on the statistics. After edits are made, the revised map would be re-positioned in the space to see how it moves and which group of experts it gets closer to. For a high dimensional space that has more than 3 features, we apply t-SNE (t-Distributed Stochastic Neighbor Embedding) [[166](#_bookmark299), [167](#_bookmark300)] to transform the multidimensional data into a two dimensional data so that we can easily plot and display the maps visually to the

student. The parameters involved here include the iteration time (*t*) and the similarity distance function (*f* ).

## Discussion and Future Work

As discussed earlier, the fields of education and network science have been exploring various methods and software to apply graph-based approaches that represent knowledge structures to assess and leverage students’ learning [[3](#_bookmark136), [55](#_bookmark188), [56](#_bookmark189), [145](#_bookmark278), [147](#_bookmark280), [148](#_bookmark281)]. However, map assessment has relied on one single expert map, and there has been less of a focus on how to handle multiple expert maps in settings such as ill-structure problems, which constitute the foundation of problem-based learning [[120](#_bookmark253), [121](#_bookmark254), [122](#_bookmark255)]. To address this gap, this chapter introduces a framework design that employs four various map comparison strategies, with a focus on handling the multiplicity of expert maps in knowledge map assessment and providing feedback derived from the comparison between maps to guide students’ learning.

### Analysis of Four Comparison Strategies

Note that our objective in the design is not to list endless combinations of strategies, but to focus on four clearly distinct approaches. These four comparison strategies take into account several important factors (e.g., methods of similarity calculation, selection of expert maps in comparison process, and feedback provided to students) for map comparison. Each of these four approaches has its own pros, cons and different applicability.

*Strategy One* is simple and straightforward to understand and implement as it continually and iteratively provides the student with positive information that helps them construct and modify their map towards the benchmark of experts. The nature of showing students with only similar expert cases makes it constrained to situations that promote confirmation with clearly recognizable patterns in problem conceptualization [[168](#_bookmark301)]. For example, in the task of summary writing, despite some variability from expert to expert in understanding, experts usually construct similar knowledge structures [[148](#_bookmark281), [169](#_bookmark302)].

For problems that admit multiple perspectives and solutions, to avoid *local optimiza- tion in thinking*, *Strategy Two* takes into effect. In this strategy, besides compatible cases, conflicting evidence are integrated in the model as many studies have shown that *cognitive disequilibrium* and its aﬀiliated *state of dissonance* are beneficial to promote deeper learning and develop critical thinking [[162](#_bookmark295), [163](#_bookmark296), [170](#_bookmark303), [171](#_bookmark304)]. However, the parameter *α* that controls the ratio of consonant and dissonant cases is hard to define. An *α* that is too low may result in insuﬀicient challenge and a high level of support, thus creating little progress in learning. Conversely, an exceedingly high *α* would result in a hard challenge with little support, hence students may struggle [[172](#_bookmark305)]. Neither of these situations promotes learning growth.

*Strategy Three* incorporates GED with another similarity calculation function so that action sequence and immediate feedback could be provided to students to help them examine how close their maps are to expert maps. Although more information is generated to assist students in modifying their maps, the computational cost also increases significantly. The

time complexity for finding the GED between a student map with *N* nodes and one expert map with *M* nodes in one iteration is *MN* via beam search algorithm [[145](#_bookmark278), [173](#_bookmark306)]; quick heuristics are now available, but they would still be called numerous times (at each step and for each expert) hence the cost would still be significant. Considering that this strategy also contains another similarity calculation function and the whole process is iterative, it would potentially be computationally prohibitive for large expert maps consisting of many nodes. Another factor to consider for applying this strategy is the misalignment between two similarity algorithms. For example, students may pick the expert map that is the most “similar” based on the selected similarity calculation function but it may be a lot more work according to the GED result to get to it than a more “dissimilar” map.

*Strategy Four* applies a cluster-based approach that groups experts maps into different clusters so that the student is able to learn how many directions it takes to solve the problem holistically. As a student can observe the moving trajectory of their map in a 2D space, the student can easily know whether he is heading in the right direction with his map modification. This approach is more applicable to cases that involves a large number of expert maps. The clustering is unnecessary and meaningless for small sets of expert maps.

### Future Work

This theoretical framework is well grounded in a comprehensive synthesis of extant schol- arship on map comparison and graph theory; thus, it offers promising value for guiding stu- dents’ map construction using multiple expert maps. The eﬀicacy of each approach should be tested on the data sets presented in other chapters (i.e., Chapter [3](#_bookmark49) and [4](#_bookmark73)) when more ex- pert maps are collected. Before implementation and experimental evaluation, several aspects should be examined further.

First, to perform knowledge map comparison, we need to identify and define a *measure of distance or difference* between maps. Grounded in graph theory and machine learning, numerous comparison methods have been proposed and implemented in current systems (e.g., HIMATT, ITACM). Yet the methods are often sensitive to the domain of application and characteristics of the data sets [[158](#_bookmark291)], so much work of experimental evaluation is still needed to decide which methods work best in which scenarios. This requires a thorough examination of the characteristics of our data sets.

Second, in all proposed methods, we set one halting condition of the iterative comparisons as “until the student decides to stop”. This setting may result in a situation in which the student chooses to stop too early as the student (erroneously) believes that suﬀicient work has been performed. One potential replacement is to set a similarity threshold, which indicates the student’s map is suﬀiciently similar to the expert maps. An algorithm is needed to detect the plateau of similarity change, otherwise the threshold may never be met and the student would be frustrated in a system that would seemingly hold them forever.

Third, in terms of specific methods, we explained graph edit distance to generate action sequences in our *Strategy Three*. A question regarding whether these personalized action sequences could be clustered is worthy of careful examination. If homogeneous groups of action sequences could be identified, showing every GED from each expert map would be

redundant and easily cause confusion to students. Further exploration into these comparison approaches would allow researchers to provide more eﬀicient approaches to assist students using expert maps for learning improvement.

## Conclusion

In this chapter, we present an algorithmic framework that combines the graph theory and map assessment to handle the multiplicity of expert maps that represent various knowledge structures and perspectives. This framework allows students and educators to select different similarity calculation functions and four distinct comparison strategies of using different expert maps to guide and leverage students’ learning. The selection of comparison strategies is based on the knowledge domain and characteristics of the expert maps.

**Chapter 6**

# Designing the Next Generation of Map Assessment Systems

Instructional strategies employing complex problem-solving examine how learners repre- sent the problem space and argue for decisions. Unlike problems with few solutions (e.g., multiple choices), instructors struggle to evaluate solutions to ill-structured problems quickly and consistently across students. This prevents instructors to provide timely feedback to learners, who may continue to engage in erroneous thinking. Despite the availability of sev- eral software packages to assess the learners’ works as maps, several barriers still prevent the wide-scale adoption of such assessment systems by instructors. We examine three open ques- tions and identify opportunities from Artificial Intelligence to develop the next generation of assessment systems.

## Introduction

Educators are increasingly exploring instructional strategies that employ problem-solving. For example, *Problem-Based Learning* (PBL) provides a problem at the start of a unit of instruction to guide the learning agenda, thus promoting collaboration among learners while shifting the instructor’s role to facilitators of learning [[174](#_bookmark307)]. PBL relies on an *ill-structured problem*, which admits multiple solutions and calls for a careful analysis of the problem space [[122](#_bookmark255)]. For example, such problems include a case in which learners must hire a can- didate, thus balancing aspects such as fairness and motivation [[114](#_bookmark247), [115](#_bookmark248)], or a case where a possible tax rebate to attract a company [[5](#_bookmark138)] invokes notions of equity, sustainability, and the labor market (Figure [6.1](#_bookmark122); left). When compared with didactic forms of instruction, such problems promote systems thinking skills in learners, as they need to represent the broader problem space and navigate it to support their decisions with arguments.

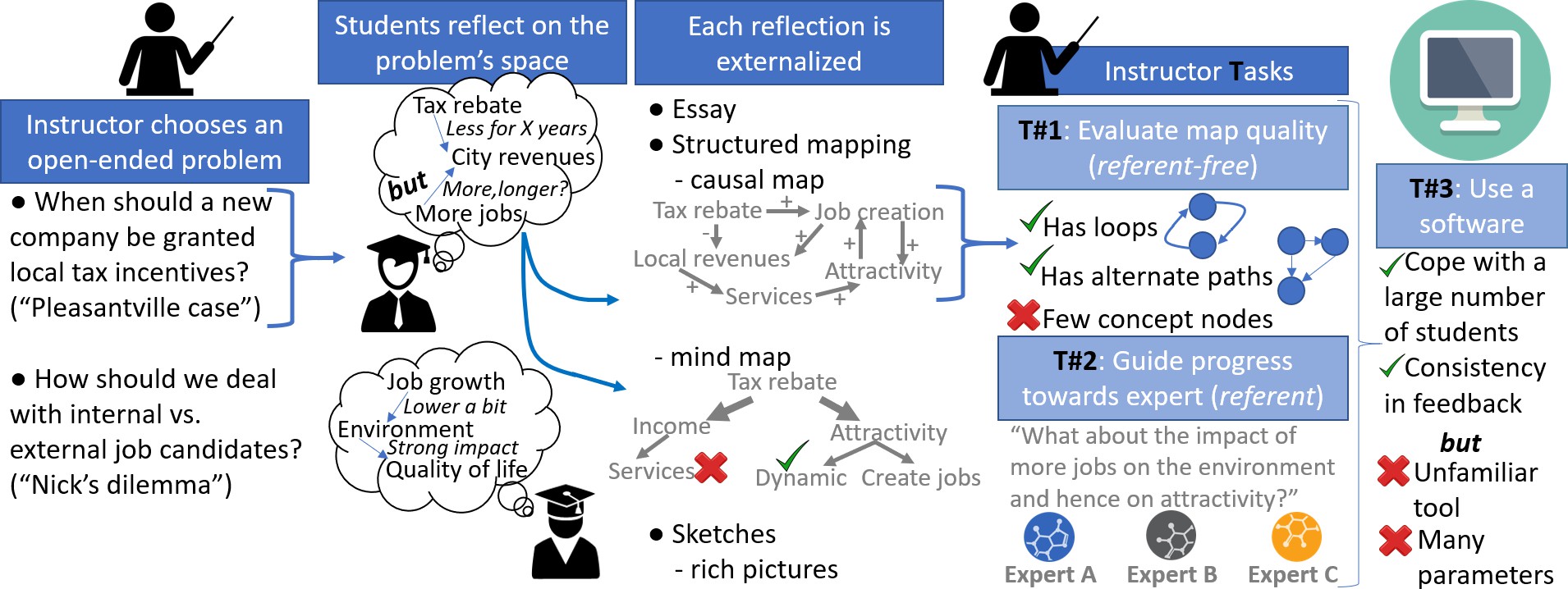


Figure 6.1: Our study focuses on three instructor-led tasks pertaining to the assessment of learners’ knowledge externalized in the form of causal maps. The three tasks cover the reference-free approach (i.e., what is a good map?), the reference-based approach (i.e., how does a novice learner differ from an expert?) and the ability to use software packages..

In contrast to problems with a prescribed set of solutions (e.g., multiple choices), instruc- tors struggle to evaluate solutions to ill-structured problems quickly and consistently across students. For example, a learner’s view of the problem space and accompanying arguments can be externalized in the form of an essay, which can be subject to mental fatigue during evaluation by an instructor [[175](#_bookmark308)]. In addition, learners can hardly be expected to produce a prescribed right answer via their complex mental model, given that multiple solutions exist. This makes it diﬀicult for instructors to provide timely feedback to learners, who may thus continue to engage in erroneous thinking. Indeed, despite the importance of causal reason- ing, it has been argued that because “there are many possible solutions to ill-structured problem-solving, as well as many possible paths to take to those solutions, assigning conven- tional letter grades may be diﬀicult” [[176](#_bookmark309), p. 39]. The National Research Council [[177](#_bookmark310)] thus

describes that a “new vision for science teaching and learning poses challenges for assessment and will require significant changes to current assessment approaches” [p. 43].

Automatic grading systems have been studied for years and several such software pack- ages have specialized in supporting instructors when assessing a learner’s perception of the problem space. Among several ways in which this internally-held perception can be external- ized for assessment (Figure [6.1](#_bookmark122); center), approaches rooted in structured mapping have often been used. These maps are more similar to expert solutions than text [[12](#_bookmark145)], thus providing a valuable way to compare the perspectives of a novice learner with an expert. Such maps are also convenient to externalize, since their emphasis on relationships closely aligns with how facts are stored in semantic memory [[178](#_bookmark311), [179](#_bookmark312)]. As a result, software packages often repre- sent a learner’s mental model as a *causal map* [[127](#_bookmark260)], either imported from mapping software (e.g., Coggle, cMap) that are increasingly recognized to promote greater systems thinking skills than paper-based modeling [[180](#_bookmark313)], or transformed from essays to obtain the underlying mental model structure (e.g., GIKS, ALA-Reader and Pathfinder KNOT). Causal maps con- sist of concepts as nodes and directed links between concepts to indicate causality, labeled by polarity (when A increases, B increases/decreases). Note that causal maps differ from Novakian *concept maps*, which are also well-studied tools to assess structural knowledge [[181](#_bookmark314)] but require “linking words” between concepts and are primarily top-down diagrams, whereas causal maps often have loops.

While this externalization may omit some of the nuances held in a learner’s mental model (e.g., strength or uncertainty of causality, conditions), it provides a well-defined mathemati- cal representation that supports several algorithms for evaluation. Indeed, multiple software such as HIMATT [[144](#_bookmark277)] or SMART [[148](#_bookmark281)] offer a variety of such algorithms to assess a map either by looking for desirable structural features (*reference-free approach*) or by contrast- ing it with an expert’s work (*reference-based approach*). Some packages also automatically provide feedback to students, thus leveraging artificial intelligence techniques from unstruc- tured (graph) data to support students in forming decisions [[60](#_bookmark193), [145](#_bookmark278)]. Empirical evidence has shown that such software present a potential to scale the evaluation to many learners in educational settings [[144](#_bookmark277)], in addition to being internally consistent due to a reliance on algorithms rather than manual evaluation.

Despite these successes, several barriers currently prevent the wide-scale adoption of map assessment systems by instructors. In this chapter, we focus on three key tasks (Fig- ure [6.1](#_bookmark122); right) and identify opportunities afforded by Artificial Intelligence (AI) to guide the development of the next generation of assessment systems. The first task is to assess the *structural quality of a learner’s map* (i.e., reference-free approach), which is one of the National Research Council recommendations to assess conceptual and structural knowledge in learners [[27](#_bookmark160)]. There is a large variety of criteria to choose from [[182](#_bookmark315), chapter 10] and each software implements several of these choices, resulting in a collection of “toolboxes” that delegate to end-users the responsibility of identifying desirable features for a learner’s map. However, *finding the right criteria* is itself a complex process (e.g., depending on the level of education or mapping tasks) and an ongoing research topic. In addition, instructors are not typically versed in network analysis, hence they may struggle to choose a network algorithm

among others and then interpret its results. This is a broad problem given that maps are often assessed, as exemplified by a recent review that reported 460 empirical studies scoring maps [[182](#_bookmark315), chapter 10]. To address this challenge and equip instructors with a *smaller set of clearly distinct measures that translate to student learning outcomes*, we present opportunities from the field of feature extraction and selection applied to causal maps in education.

The second task is to guide a student in producing a map that *increasingly incorporates elements of an expert solution* [[4](#_bookmark137), [52](#_bookmark185)]. Currently, instructors using this reference-based approach can only use one expert map at a time. That is, they need to decide which expert is “right”, although *ill-structured cases admit multiple solutions*. To address this long-standing misalignment, we explore possible ways through which artificial intelligence techniques can compare a student’s work with multiple experts, thus combining aspects of human intelligence with AI to guide a student without constraining their viewpoint.

Finally, there is a growing recognition that AI solutions for knowledge assessment will not “take over” the role of teacher [[183](#_bookmark316)]; rather, we view the *instructor as a key end user of technology so they can maximize complex problem-solving and deep learning*. However, current software has largely been developed without a design framework grounded on the needs and practices of instructors. A review of assessment for computer-based concept maps (which are closely related to the concept maps of interest here) previously noted that only 24 out of 119 studies discussed usability [[184](#_bookmark317)], thus highlighting this disconnect between creating elaborate systems via AI and whether people could adequately use them. Consequently, conceptualizing the *use of a software as its own task* allows us to emphasize usability, which has so far limited the uptake of AI advancements within classrooms. Building on the recent work of Rohles [[182](#_bookmark315), chapter 11 & 12], we explain how the next generation of map assessment systems can be designed more closely with its intended user base by developing design principles.

## First Task: Finding Suﬀicient Criteria to Evaluate the Structural Quality of a Learner’s Map

The structure of a learner’s causal map is a graph, in which the nodes have labels in- dicating their meaning and edges have a type (positive or negative causation) as well as a directionality. Algorithms from the field of network science can thus be applied to this directed, labeled, weighted graph to extract characteristics of interest. These characteristics fall into two groups. At the *graph-level*, each characteristic produces a single number based on the entire map. Six such characteristics are illustrated in Figure [6.2](#_bookmark124). For example, the diameter can be interpreted as the maximum distance between two concepts. If the diameter is exceedingly large, a learner may be going on a tangent, or overlooked connections between existing concepts (which would create shortcuts and hence lower the diameter); instructors may thus apply penalties for maps based on their diameters. At the *node-level*, a number is associated to each node to represent its *importance or centrality*. For instance, we can count how many concepts are directly impacted by a node (i.e., the out-degree). A map

with a single central concept (e.g., one notion that impacts or is impacted by everything else) would be indicative of an early work, which has not yet comprehensively examined the problem space. In contrast, an expert map has a few central elements which serve as focal points, and selective concepts (i.e., few nodes) but highly intertwined [[185](#_bookmark318)].

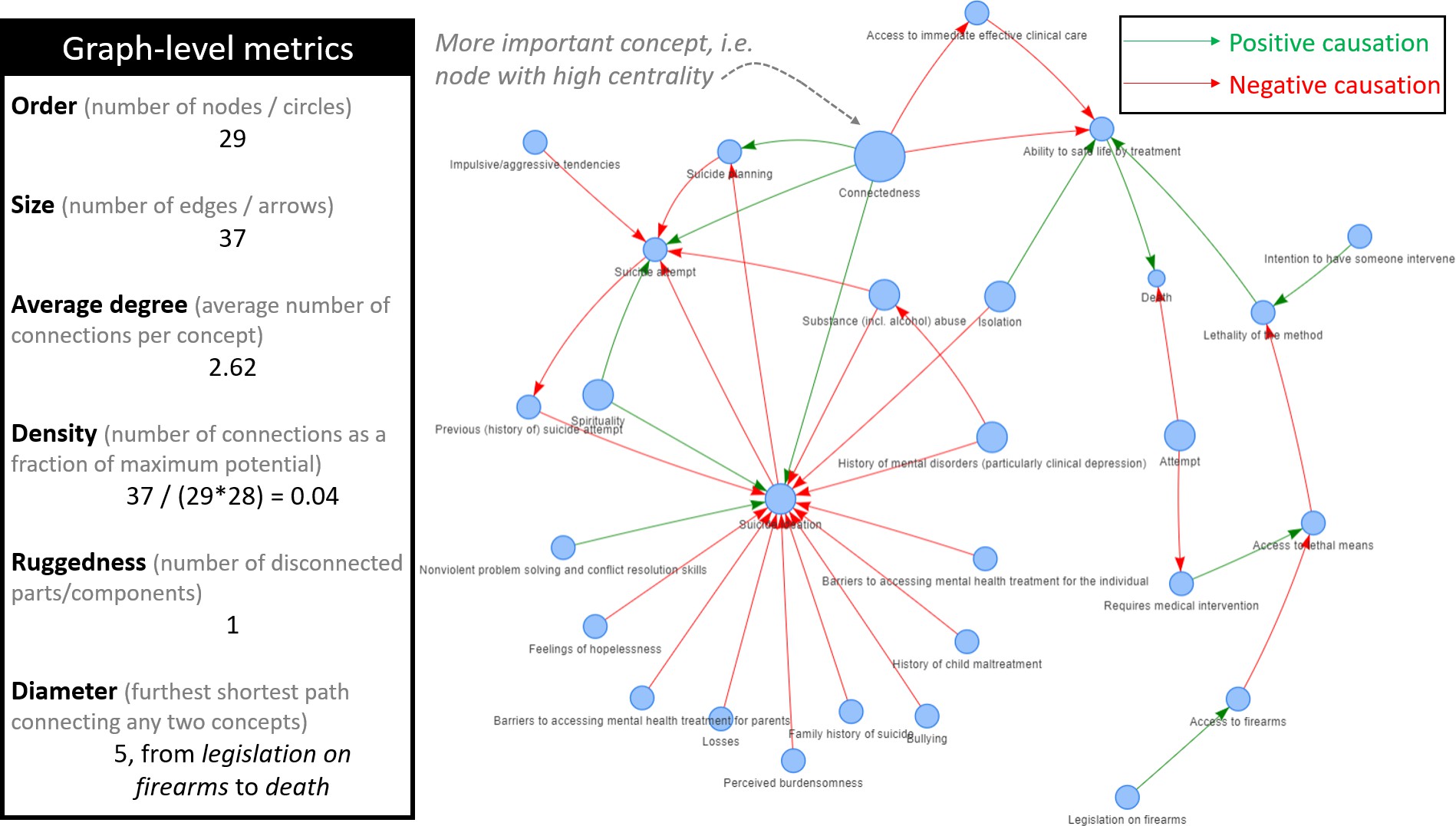


Figure 6.2: A map can be assessed via graph-level metrics, each producing a single number based on the entire map. A map can also be assessed via node-level metrics, where each node is associated with a number. Some nodes may have the same numbers (shown here as nodes of identical sizes).

When pooled together, previous studies have used over ten different graph-level metrics [[4](#_bookmark137), [5](#_bookmark138), [7](#_bookmark140), [46](#_bookmark179)] as well as many node-level metrics [[11](#_bookmark144)]. This wide array of choices is a problem for instructors for at least three reasons. First, relating each one to the *quality* of a map can be challenging [[77](#_bookmark210)], hence instructors may *struggle in effectively using the numbers*. Second, several of these measures are *highly-correlated* [[186](#_bookmark319), [187](#_bookmark320)], hence instructors may be wasting time analyzing results without realizing that they are essentially the same, and/or guiding students by *over-emphasizing* aspects which are repeatedly counted through several measures. Third, some of these measures are *computationally costly* to obtain [[188](#_bookmark321)], which can become a problem when evaluating many maps obtained in classes with large enrollment. It is thus essential to reduce the set of metrics to those that provide complementary rather than redundant perspectives.

We posit that the Machine Learning (ML) technique of *feature selection* holds the key to identifying suﬀicient structural assessment criteria. This technique is used for dimensionality reduction, as it automatically finds the most relevant set of features in a data set [[189](#_bookmark322), [94](#_bookmark227)].

We would thus need empirical studies that apply feature selection methods on their maps, in order to evaluate assessment for future maps. The analysis can follow a three steps process (Figure [6.3](#_bookmark125)). First, a data spreadsheet must be produced for ML, consisting of individual maps as rows and a large set of metrics as columns. Then, several feature selection algorithms would be applied to rank features by their importance. It is indeed customary in applied ML research to employ several algorithms, since each one is adept at finding different types of patterns in the data. The last step is to pool their results by computing the average rank of each feature. Such a data-centric approach would suggest which features to eliminate (with a low-rank) and which ones should the focus of guidelines for interpretation (with a high-rank) as well as a standard for assessment. Such empirical ML studies would need to be *performed on several mapping tasks and with learners of varying characteristics*, otherwise the features that are statistically relevant for one context may not aptly serve the needs of instructors in another setting. Since the same analysis should be performed across data sets, the field would benefit from a reusable and openly accessible script. Otherwise, differences between results may be caused by the use of different algorithms or computer codes, or even the possibility of scripting errors.

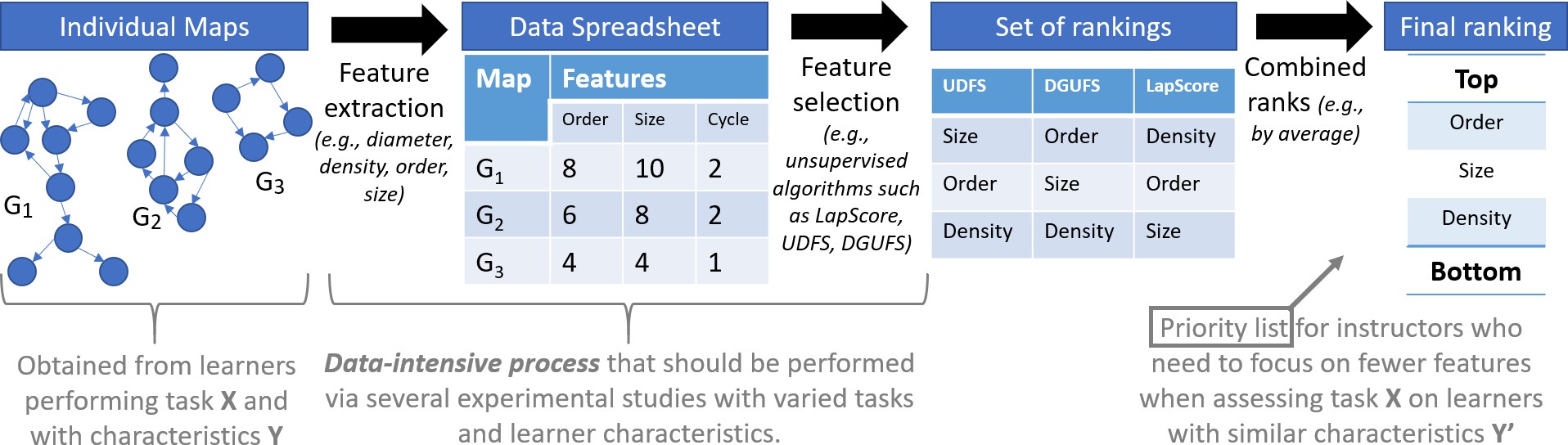


Figure 6.3: By performing feature extraction and feature selection across maps collected from learners of various profiles and working on diverse tasks, we can identify the most important metrics and thus guide the assessment effort of instructors for similar tasks and learners.

## Second Task: Guiding a Learner in Progressing To- wards a Multiplicity of Experts

While the previous task can serve to answer the summative question “is it a good map?”, instructors also need formative assessment techniques to help learners improve their maps [[179](#_bookmark312)]. The transition from a summative aspect to a formative focus can be a challenge if an instructor only employs the metrics discussed in the previous section. For example, a learner may have a penalty for having too few concept nodes, but “think of more nodes” is not an easily actionable feedback. Expert maps thus serve to provide specific feedback, as the contrast between a learner (or “novice”) and an expert can reveal *specific* differences [[3](#_bookmark136)],

leading to more actionable and convincing feedback such as “an expert saw environmental pollution as part of the problem space –where do you think that it would fit in your map?” A shortcoming of previous environments for expert-novice knowledge transfer is their com- parison of the learners’ maps with *a single expert map at a time*, which forces learners into espousing a specific worldview despite the open-ended nature of ill-structured problems. Sev- eral domains have shown that experts do not necessarily think alike. As an example, studies in health services that applied mapping tools to externalize expert perspectives found that experts *held different views on the same problem*, depending on factors such as their area of specialty (e.g., policymakers or primary care practitioners in [[190](#_bookmark323)] or as many as six disci- plines in [[191](#_bookmark324)]), their level of proximity with a patient [[192](#_bookmark325)], or cultural factors [[193](#_bookmark326)]. It is thus necessary to create learning environments that effectively leverage a multiplicity of ex- pert maps. This leads to two related matters of selection and integration: which expert maps should be selected to promote knowledge gains in the learner? And as multiple maps are selected, how do we provide guidance to *integrate* their (potentially conflicting) viewpoints into the learner map?

Technically, it is possible to select the expert maps that most closely resemble the learner map, ask the learner to consider some of their facts to update the map, and then re-compute which expert maps are closer as an iterative process (Figure [6.4](#_bookmark127)). However, this can lead to an “echo chamber” and promote limited growth. In contrast, previous research has recom- mended to prompt knowledge change through a learning environment designed to “induce puzzlement or *cognitive conflict* by carefully introducing a set of facts that are at first con- tradictory to what which the learner believes” [[194](#_bookmark327)]. The theory of cognitive dissonance, originating in social psychology [[162](#_bookmark295)], posits that nodes in the learner map are then either in conflict (i.e., dissonant) or compatible (i.e., consonant) with the evidence. Conflicts are uncomfortable for learners, mentally and even physically [[163](#_bookmark296)], hence they feel compelled to react. This reaction can be as simple as ignoring the evidence or prompt deeper changes in the map by adding new nodes and/or changing the weight of existing connections. For example, a learner advocating against a smoking ban may be exposed to evidence on the neg- ative health effects of smoking, to which a reaction would be to add nodes on how smoking may reduce tension and prevent weight gain [[195](#_bookmark328)]. Our experiments have provided empirical evidence that learners tend to grow their maps most when exposed to a case study depicting a (surprising) failure, compared to learners exposed to a case study confirming their initial views [[5](#_bookmark138)].

In terms of integration, a core notion for knowledge transfer is *resituation*, that is, the

ability of a learner to adapt knowledge from a different context into their own [[196](#_bookmark329)]. It is relatively straightforward to follow this notion when designing a model-based learning envi- ronment that admits a single expert. Indeed, a learner’s model can be compared with the expert and a sequence of steps is then provided to guide the learner in gradually bridging the gap with the expert map [[3](#_bookmark136)]. In contrast, the situation can become disorienting when multiple expert maps are used, as a learner may be advised to take a sequence of actions that contradict each other or seem unrelated. In addition, exposing a learner exclusively to con- flicting evidence and hence causing a repeated discomfort may not constitute an environment

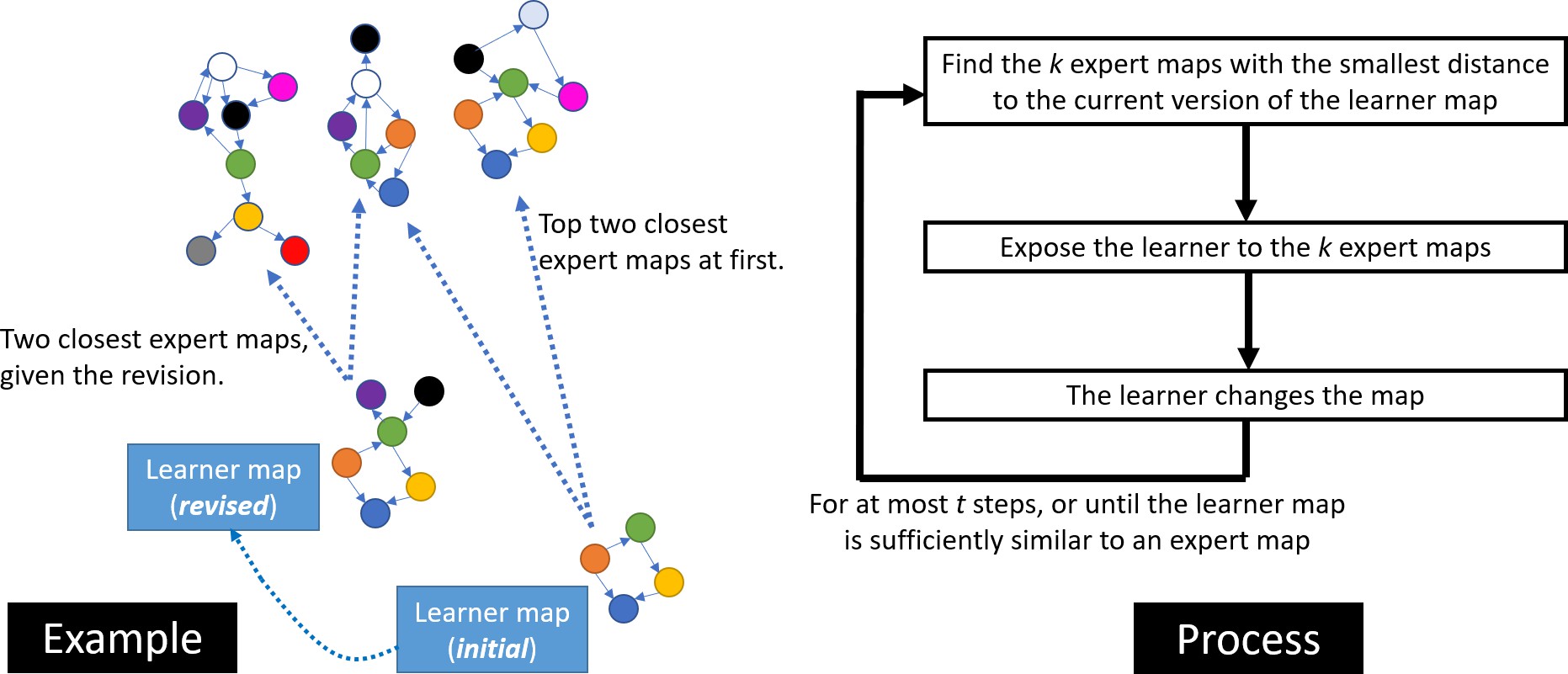


Figure 6.4: A learner can be selectively exposed to the most similar expert maps and make revisions. The revised map may drift closer to a different subset of experts hence the distance would be recomputed. The notion of similarity depends on the distance metric between maps.

conducive to learning. There is evidence “for the value of *multiple iterations of challenge- support* phases to develop increasingly robust understanding over time” [[153](#_bookmark286)], which suggest that learners can benefit from a cycle of challenges by exposure to very different map and then receiving support to address this dissonance.

A running thread when handling a multiplicity of experts is to *measure the distance* or “difference” between maps. Similarly to the previous task, there is an abundance of such measures, and each one can emphasize different aspects. A simple approach considers that two maps are more similar if they share many connections (Figure [6.4](#_bookmark127)), which ignores the weight of these connections and assumes that concept nodes have clearly matching names. The cognitive distance between maps introduced by Aminpour, Schwermer, and Gray goes one step further by accounting for the weight of nodes [[159](#_bookmark292)]. The most comprehensive comparison framework is provided by the Graph Edit Distance [[145](#_bookmark278)] as it identifies the minimal sequence of changes to turn one map into another, thus accounting for differences in edge weights, missing/excess edges or nodes, and differences in node names. Note that *none of these distance metrics were established based on pedagogical criteria*, hence there may be a disconnect between a distance and the effort required from the learner. That is, a map with a small distance may prompt more work than a map whose distance scores higher. Three criteria have recently been proposed for the similarity of *concept maps* based on pedagogical considerations [[197](#_bookmark330), [198](#_bookmark331)], but they are rooted in the hierarchical structure of concept maps whereas causal maps can have cycles. At present, there is thus an ongoing need to develop similarity measures for causal maps that adequately correlate with a learner’s effort, as well as experimental studies to evaluate the effect of different measures.

## Third Task: Improving Software Usability via a Practical Design Framework

A key aspect of technology design includes the interface design and related human- computer interaction. Although educators must provide targeted feedback on students’ reasoning [[176](#_bookmark309)], the field has *yet to establish a teacher-centric design theory for these auto- mated assessment technologies* [[199](#_bookmark332), [200](#_bookmark333)]. As theorists and practitioners explore automated assessments, additional insight is needed around design principles to ensure that novel AI tools constitute effective resources for educators. To date, diverse theories and models have explored various problem-solving holistically, such as the ones outlined by Jonassen [[201](#_bookmark334)] or the 4C I/D model [[202](#_bookmark335)]. Others outline specifics in areas such as question generation, argumentation, or information literacy. However, few have explored these AI tools from a human-computer interaction perspective. As noted by Novak et al. [[203](#_bookmark336)], “despite a growing body of research in the area of digital learning and information processing, the literature on how people process and interact with information on electronic devices and computers is still very scarce” [p. 151].

One reason for the lack of clarity regarding the role of human-computer interaction and learning technologies relates to the methods used to explicate the phenomenon. Broadly speaking, HCI applied to learning technology can be described as the interactions that users (teacher, students) engage towards the process of learning. This aspect is unique because it diverges in some respects from other views of HCI. While others are focused on designs that are eﬀicient and seamless, the HCI of learning technology is also concerned with processes such as cognitive disequilibrium. The HCI component may inherently include interfaces that challenge the learner, while simultaneously designing core components that are seamless (e.g., navigation). Special consideration is thus needed to simultaneously design for the interaction with the learning space (learning actions) and interaction with the learning environment (technology) [[204](#_bookmark337)].

A HCI view of learning technologies includes important implications for the design of automated assessment tools, especially in terms of methodology. To date, many post-hoc learning outcomes have been explored in terms of self-report surveys, test scores, or learning artifacts (e.g., argumentation essays). To better understand the design principles needed for automated assessments software, the field should engage a more diverse array of methodolo- gies and data collection approaches (Figure [6.5](#_bookmark129)). For example, scenario and persona activities allow designers to considers users’ broader socio-technical context in which the learning tech- nology is situated. Card sorting is often used to understand how users group core features together, which may explicate how users conceptualize their goal-directed behaviors during the learning activity. Alternatively, a cognitive task analysis might yield important insights into the *in situ* processes of users and their initial orientation and necessary interface features needed to complete the task [[205](#_bookmark338)]. The application of these HCI methodologies may help the field move towards user-centered design principles for learning technologies that support automated assessments for deep learning.

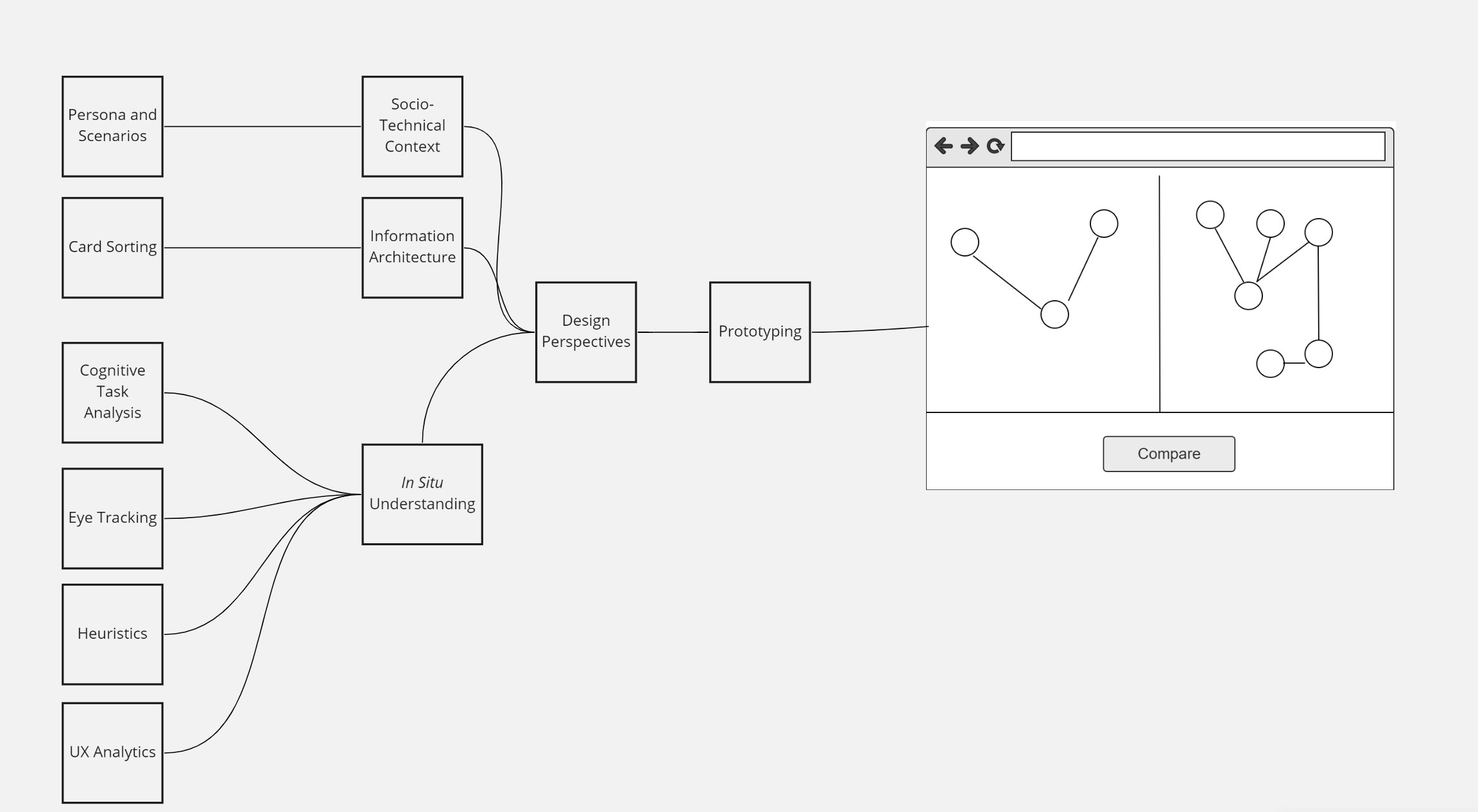


Figure 6.5: Human-Computer Interaction as informing design principles for learning.

## Discussion

As education moves towards a more ill-structured approach, theorists have advocated for alternative ways for learners to represent complex problem-solving and achieve deep learning. Instead of asking learners to provide simple responses that signify rote memorization, map- ping approaches have a rich history in allowing learners to externalize their mental models, thus creating artifacts that can be automatically examined via several software [[144](#_bookmark277), [127](#_bookmark260)]. Automatization is essential to support instructors in consistently and quickly performing summative or formative assessments of student learning through maps. However, there is still a notable divide between the emergence of AI solutions and their uptake for assessment. To address this gap, we examined three tasks that instructors need to achieve: summative assessment of a learner map, formative assessment using both a learner map and expert maps, and effectively using software for assessment. We examined specific challenges fac- ing each of these tasks and proposed specific methods to address these challenges. This paper thus supports conceptual conversations on assessment by mapping and also provides practical steps to fuel research and development regarding the next generation of mapping assessment software.

Although we focused on three tasks, we acknowledge that gaps between software and practices for map-based assessment include other tasks and hence additional limitations. In particular, Ruppert, Duncan, and Chinn [[206](#_bookmark339)] emphasized the role of domain-specific knowledge, by showing that modeling strategies did not suﬀice to develop a strong causal map and that “scaffolding certain *domain-specific knowledge* is necessary to assist students

with incorporating evidence in modeling tasks.” Current mapping software are generally limited to the maps and aim to achieve knowledge transfer by exposing a learner to (specific elements of) expert maps; this approach provides some domain-specific knowledge but falls shorts of having learners collect data or examine scientific studies. There is thus a potential to incorporate case libraries as part of mapping software to ensure that learners can be prompted to examine repositories of domain-specific knowledge that go beyond just using maps.

**Chapter 7**

# Conclusion

Knowledge maps have been widely applied in knowledge elicitation and representation to evaluate students’ learning and their knowledge structures, particularly in problem-based learning contexts. To effectively construct a knowledge map and use it as an evaluation tool, several factors (e.g., map features, structure of map, and nature of group interaction) and multiple expert maps as guidance should be carefully taken into account. Through- out this thesis, we conducted studies that examine these aspects computationally and offer implications on advancing the application of knowledge maps in student learning.

## Contributions

Overall, this thesis makes contributions to the research fields of network science and edu- cation by investigating multiple computational approaches to construct and assess knowledge maps. Specifically, through analyzing graph features of 202 knowledge maps from four differ- ent case studies via unsupervised feature selection techniques, this work provides promising guidance and reveals how and which map metrics could be selected in future studies of knowledge map assessment. In addition, by investigating the impact of group interaction and group size via analyzing 54 knowledge maps, implications are offered to instructors and scholars to help them implement and assess map construction in the classroom. Further- more, the proposed algorithmic framework of examining and integrating multiple viewpoints reflected in the expert maps provides insights and promising techniques of incorporating a variety of perspectives in guiding students’ learning, such as iteratively providing feedback to guide students in constructing maps via map comparison in multiple ways.

Model-based learning environments take a cognitive approach to educational technology. Despite their potential, such environments also have several limitations that contribute to preventing their wider use by instructors. The summative assessment of a learner’s map is a challenge for instructors who are faced with a myriad of measures, which can be diﬀicult to select, compute, and interpret. In this thesis, we proposed concrete directions for research that leverage Artificial Intelligence (specifically feature selection techniques within machine learning) to enhance the human intelligence of instructors performing a summative assess- ment. We also examined the matter of formative assessment through the lens of knowledge transfer from experts to (novice) learners. While this subject has been extensively researched previously, there is still a dearth of solutions when problems admit multiple yet conflicting expert perspectives. Transferring the human intelligence of experts to learners can be fa- cilitated by another set of Artificial Intelligence techniques, so we articulated opportunities

for future research to leverage cognitive distances as a means to create learning environ- ments that promote a diversity of worldviews. Finally, we note that a successful interaction between AI and human intelligence necessitates a usable software environment. We thus explored how usability and the identification of design principles are essential goals if future environments for educational technology aim to create sustained and beneficial interactions between AI and human intelligence.

## Limitations and Future work

Our research relied heavily on rapidly evolving techniques and theories, such as machine learning, feature selection, and network comparison, and there are numerous algorithms, tools, and designs in development. As a result, we might not have selected and used the most optimal strategy for every aspect of our research. For example, in Chapter [3](#_bookmark49), we only covered filter and wrapper approaches of unsupervised feature selection (UFS) and did not examine any hybrid types that integrate the advantages of both filter and wrapper methods [[97](#_bookmark230), [92](#_bookmark225)]. One direction of future work could be to explore hybrid methods of UFS to examine whether the ranking of the importance for graph features would change.

Another limitation that we faced is collecting data of high quality. To effectively reflect students’ knowledge structure via knowledge maps, it is essential for mappers to master some important mapping skills. For example, a skill like *articulation* is essential to help the student represent distinctly the key concepts and important links between concepts [[17](#_bookmark150)]. Given that our data sets were collected through different venues, the mappers received limited suﬀicient training on mapping prior to constructing their maps on some occasions. This limited our analysis of maps to those constructed in settings with minimal prior training.

Furthermore, a technical challenge regarding linguistic variability (see Section [2.4.3](#_bookmark43)) re- mained in our research. Our study mostly focused on the graph structure instead of the content. For instance, the algorithmic framework we designed for handling the multiplicity of expert maps is built on the prerequisite that semantic inconsistency is resolved before map comparison. Therefore, future work regarding a full study of terminology alignment in the context of graph comparison is needed.

## References

1. David L Trumpower, Mehmet Filiz, and Gul Shahzad Sarwar. Assessment for learning using digital knowledge maps. In Dirk Ifenthaler and Ria Hanewald, editors, *Digital knowledge maps in education*, pages 221–237. Springer, 2014.
2. Joseph D Novak and Alberto J Cañas. The theory underlying concept maps and how to construct and use them. *Florida Institute for Human and Machine Cognition*, 1(1):1–31, 2006.
3. Philippe J Giabbanelli and Andrew A Tawfik. Reducing the gap between the con- ceptual models of students and experts using graph-based adaptive instructional sys- tems. In *International Conference on Human-Computer Interaction*, pages 538–556. Springer, 2020.
4. Heiko Krabbe. Digital concept mapping for formative assessment. In Dirk Ifen- thaler and Ria Hanewald, editors, *Digital knowledge maps in education*, pages 275–297. Springer, 2014.
5. Philippe J Giabbanelli and Andrew A Tawfik. How perspectives of a system change based on exposure to positive or negative evidence. *Systems*, 9(2):23, 2021.
6. Po-Han Wu, Gwo-Jen Hwang, Marcelo Milrad, Hui-Ru Ke, and Yueh-Min Huang. An innovative concept map approach for improving students’ learning performance with an instant feedback mechanism. *British Journal of Educational Technology*, 43(2):217– 232, 2012.
7. Dirk Ifenthaler. Relational, structural, and semantic analysis of graphical representa- tions and concept maps. *Educational technology research and development*, 58(1):81–97, 2010.
8. Kyung Kim. Graphical interface of knowledge structure: A web-based research tool for representing knowledge structure in text. *Technology, Knowledge and Learning*, 24(1):89–95, 2019.
9. Howard E Herl, HF O’Neil Jr, Gregory KWK Chung, and John Schacter. Reliabil- ity and validity of a computer-based knowledge mapping system to measure content understanding. *Computers in Human Behavior*, 15(3-4):315–333, 1999.
10. Dirk Ifenthaler, Iskandaria Masduki, and Norbert M Seel. The mystery of cognitive structure and how we can detect it: tracking the development of cognitive structures over time. *Instructional Science*, 39(1):41–61, 2011.
11. Eric A Lavin, Philippe J Giabbanelli, Andrew T Stefanik, Steven A Gray, and Robert Arlinghaus. Should we simulate mental models to assess whether they agree? In *Proceedings of the Annual Simulation Symposium*, pages 1–12, 2018.
12. Dirk Ifenthaler. Bridging the gap between expert-novice differences: The model-based feedback approach. *Journal of Research on Technology in Education*, 43(2):103–117, 2010.
13. Vishrant K Gupta, Philippe J Giabbanelli, and Andrew A Tawfik. An online en- vironment to compare students’ and expert solutions to ill-structured problems. In *International Conference on Learning and Collaboration Technologies*, pages 286–307. Springer, 2018.
14. Jeanne Ellis Ormrod and Kevin M Davis. *Human learning*. Merrill London, 2004.
15. Andrew A Tawfik and Janet L Kolodner. Systematizing scaffolding for problem-based learning: A view from case-based reasoning. *Interdisciplinary Journal of Problem- Based Learning*, 10(1):6–14, 2016.
16. Michael D McNeese and Phillip J Ayoub. Concept mapping in the analysis and design of cognitive systems: a historical review. In Brian M Moon, Robert R Hoffman, Joseph Novak, and Alberto Canas, editors, *Applied concept mapping: Capturing, analyzing, and organizing knowledge*, chapter 3, pages 47–46. CRC Press, 2011.
17. Biran M Moon, Robert R Hoffman, Thomas C Eskridge, and John W Coffey. Skills in applied concept mapping. In Brian Moon, Robert R Hoffman, Joseph Novak, and Alberto Canas, editors, *Applied concept mapping: Capturing, analyzing, and organizing knowledge*, chapter 2, pages 23–46. CRC Press, 2011.
18. Joseph D Novak and Dismas Musonda. A twelve-year longitudinal study of science concept learning. *American educational research journal*, 28(1):117–153, 1991.
19. Dawn Marie Trehan. *The impact of concept mapping as a learning tool on student perceptions of and experiences with introductory statistics*. PhD thesis, Kent State University, 2015.
20. Bilash Kanti Bala, Fatimah Mohamed Arshad, and Kusairi Mohd Noh. *System dy- namics: Modelling and simulation*. Springer, 2016.
21. Renata Jagustović, Robert B Zougmoré, Aad Kessler, Coen J Ritsema, Saskia Keesstra, and Martin Reynolds. Contribution of systems thinking and complex adap- tive system attributes to sustainable food production: Example from a climate-smart village. *Agricultural systems*, 171:65–75, 2019.
22. Team Tip. Guidelines for drawing causal loop diagrams. *The Systems Thinker*, 22(1):5– 7, 2011.
23. Kun Chang Lee and Soonjae Kwon. Online shopping recommendation mechanism and its influence on consumer decisions and behaviors: A causal map approach. *Expert Systems with Applications*, 35(4):1567–1574, 2008.
24. Kay M Nelson, Sucheta Nadkarni, VK Narayanan, and Mehdi Ghods. Understanding software operations support expertise: A revealed causal mapping approach. *MIS Quarterly*, 24(3):475–507, 2000.
25. Hendra Sandhi Firmansyah, Suhono H Supangkat, Arry A Arman, and Philippe J Giabbanelli. Identifying the components and interrelationships of smart cities in in- donesia: Supporting policymaking via fuzzy cognitive systems. *IEEE Access*, 7:46136– 46151, 2019.
26. Steven A Gray, Stefan Gray, Linda J Cox, and Sarah Henly-Shepard. Mental modeler: a fuzzy-logic cognitive mapping modeling tool for adaptive environmental management. In *2013 46th Hawaii International Conference on System Sciences*, pages 965–973. IEEE, 2013.
27. National Research Council. *The assessment of science meets the science of assessment: Summary of a workshop*. National Academies Press, 1999.
28. Leah J Cook. *Using Concept Maps to Monitor Knowledge Structure Changes in a Science Classroom*. PhD thesis, Western Michigan University, 2017.
29. Wynne Harlen. Teachers’ summative practices and assessment for learning–tensions and synergies. *Curriculum Journal*, 16(2):207–223, 2005.
30. Gregory J Cizek. An introduction to formative assessment: History, characteristics, and challenges. In Heidi Andrade and Gregory J Cizek, editors, *Handbook of formative assessment*, pages 3–17. Routledge, 2010.
31. Lorrie A Shepard. Commentary: Evaluating the validity of formative and interim assessment. *Educational Measurement: Issues and Practice*, 28(3):32–37, 2009.
32. National Research Council. *How people learn: Brain, mind, experience, and school: Expanded edition*. National Academies Press, 2000.
33. Paul Black, Robert McCormick, Mary James, and David Pedder. Learning how to learn and assessment for learning: A theoretical inquiry. *Research Papers in Education*, 21(2):119–132, 2006.
34. Carol Boston. The concept of formative assessment. *Practical Assessment, Research, and Evaluation*, 8(1):9, 2002.
35. Graham Gibbs and Claire Simpson. Conditions under which assessment supports students’ learning. *Learning and teaching in higher education*, 5(1):3–31, 2005.
36. Berit S Haug and Marianne Ødegaard. Formative assessment and teachers’ sensitivity to student responses. *International Journal of Science Education*, 37(4):629–654, 2015.
37. Philippe J Giabbanelli and Andrew A Tawfik. Overcoming the pbl assessment chal- lenge: Design and development of the incremental thesaurus for assessing causal maps (itacm). *Technology, Knowledge and Learning*, 24(2):161–168, 2019.
38. Philippe J Giabbanelli, Andrew A Tawfik, and Vishrant K Gupta. Learning analytics to support teachers’ assessment of problem solving: A novel application for machine learning and graph algorithms. In Dirk Ifenthaler, Dana-Kristin Mah, and Jane Yin- Kim Yau, editors, *Utilizing Learning Analytics to Support Study Success*, pages 175–

199. Springer International Publishing, 2019.

1. Ian M Kinchin. Concept mapping in biology. *Journal of biological education*, 34(2):61– 68, 2000.
2. Yue Yin, Jim Vanides, Maria Araceli Ruiz-Primo, Carlos C Ayala, and Richard J Shavelson. Comparison of two concept-mapping techniques: Implications for scoring, interpretation, and use. *Journal of Research in Science Teaching: The Oﬀicial Journal of the National Association for Research in Science Teaching*, 42(2):166–184, 2005.
3. Alexander Whitelock-Wainwright, Nathan Laan, Dunwei Wen, and Dragan Gašević. Exploring student information problem solving behaviour using fine-grained concept map and search tool data. *Computers & Education*, 145:103731–103749, 2020.
4. Eira Wyn Patterson. Structuring the composition process in scientific writing. *Inter- national journal of science education*, 23(1):1–16, 2001.
5. Joseph Novak. Clarify with concept maps. *The science teacher*, 58(7):44, 1991.
6. Dirk Ifenthaler. Determining the effectiveness of prompts for self-regulated learning in problem-solving scenarios. *Journal of Educational Technology & Society*, 15(1):38–52, 2012.
7. David L Trumpower and Timothy E Goldsmith. Structural enhancement of learning.

*Contemporary Educational Psychology*, 29(4):426–446, 2004.

1. Maija Strautmane. Concept map-based knowledge assessment tasks and their scoring criteria: An overview. In *Proceedings of the 5th international conference on concept mapping*, pages 80–88, 2012.
2. Mehmet Buldu and Nihal Buldu. Concept mapping as a formative assessment in college classrooms: Measuring usefulness and student satisfaction. *Procedia-Social and Behavioral Sciences*, 2(2):2099–2104, 2010.
3. Klaus Hasemann and Helen Mansfield. Concept mapping in research on mathematical knowledge development: Background, methods, findings and conclusions. *Educational studies in mathematics*, 29(1):45–72, 1995.
4. Maria Araceli Ruiz-Primo and Richard J Shavelson. Problems and issues in the use of concept maps in science assessment. *Journal of Research in Science Teaching: The Oﬀicial Journal of the National Association for Research in Science Teaching*, 33(6):569–600, 1996.
5. Evangelia Gouli, Agoritsa Gogoulou, and Maria Grigoriadou. A coherent and inte- grated framework using concept maps for various educational assessment functions. *Journal of Information Technology Education: Research*, 2(1):215–240, 2003.
6. Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
7. Dirk Ifenthaler. Scope of graphical indices in educational diagnostics. In Dirk Ifen- thaler, Pablo Pirnay-Dummer, and Norbert M Seel, editors, *Computer-based diagnos- tics and systematic analysis of knowledge*, pages 213–234. Springer, 2010.
8. Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.
9. Dirk Ifenthaler, Pablo Pirnay-Dummer, and Norbert M Seel. *Computer-based diagnos- tics and systematic analysis of knowledge*. Springer, 2010.
10. Kyung Kim, Roy B Clarianay, and Yanghee Kim. Automatic representation of knowl- edge structure: Enhancing learning through knowledge structure reflection in an online course. *Educational Technology Research and Development*, 67(1):105–122, 2019.
11. Roy B Clariana and Ravinder Koul. A computer-based approach for translating text into concept map-like representations. In *Proceedings of the first international confer- ence on concept mapping*, pages 14–17, 2004.
12. Roger W Schvaneveldt. *Pathfinder associative networks: Studies in knowledge organi- zation.* Ablex Publishing, 1990.
13. Roger W Schvaneveldt, Francis T Durso, Timothy E Goldsmith, Timothy J Breen, Nancy M Cooke, Richard G Tucker, and Joseph C De Maio. Measuring the structure of expertise. *International journal of man-machine studies*, 23(6):699–728, 1985.
14. Andrew A Tawfik, Victor Law, Xun Ge, Wanli Xing, and Kyung Kim. The effect of sustained vs. faded scaffolding on students’ argumentation in ill-structured problem solving. *Computers in Human Behavior*, 87:436–449, 2018.
15. Jorge Villalon and Rafael A Calvo. Concept maps as cognitive visualizations of writing assignments. *Journal of Educational Technology & Society*, 14(3):16–27, 2011.
16. Venkata Sai Pillutla and Philippe J Giabbanelli. Iterative generation of insight from text collections through mutually reinforcing visualizations and fuzzy cognitive maps. *Applied Soft Computing*, 76:459–472, 2019.
17. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer, 2007.
18. Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176, 2011.
19. Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949– 971, 2015.
20. Xiulei Liu, Qiang Tong, Xuhong Liu, and Zhihui Qin. Ontology matching: state of the art, future challenges and thinking based on utilized information. *IEEE Access*, 2021.
21. Peng Wang and Baowen Xu. LILY: the results for the ontology alignment contest OAEI 2007. In *Proceedings of the 2nd International Workshop on Ontology Matching*, pages 179–187, 2007.
22. Peng Wang. Lily results on SEALS platform for OAEI 2011. In *Proceedings of 6th International Workshop on Ontology Matching*, pages 156–162, 2011.
23. Bahareh Behkamal, Mahmoud Naghibzadeh, and Reza Askari Moghadam. Using pat- tern detection techniques and refactoring to improve the performance of asmov. In *2010 5th International Symposium on Telecommunications*, pages 979–984. IEEE, 2010.
24. Yves R Jean-Mary and Mansur R Kabuka. ASMOV: results for OAEI 2008. In *Proceedings of the 3rd International Workshop on Ontology Matching*, volume 431, pages 132–139, 2008.
25. Xiao Zhang, Qian Zhong, Juanzi Li, Jie Tang, Guotong Xie, and Hanyu Li. RiMOM results for OAEI 2008. In *Proceedings of the 3rd International Workshop on Ontology Matching*, pages 182–189, 2008.
26. Zhichun Wang, Xiao Zhang, Lei Hou, Yue Zhao, Juanzi Li, Yu Qi, and Jie Tang. RiMOM results for OAEI 2010. *Ontology Matching*, 195, 2010.
27. Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. RiMOM: A dynamic multistrategy ontol- ogy alignment framework. *IEEE Transactions on Knowledge and data Engineering*, 21(8):1218–1232, 2008.
28. Sivan Albagli, Rachel Ben-Eliyahu-Zohary, and Solomon E Shimony. Markov network based ontology matching. *Journal of Computer and System Sciences*, 78(1):105–118, 2012.
29. Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-match: an algorithm and an implementation of semantic matching. In *European semantic web symposium*, pages 61–75, 2004.
30. Fausto Giunchiglia, Aliaksandr Autayeu, and Juan Pane. S-match: an open source framework for matching lightweight ontologies. *Semantic Web*, 3(3):307–317, 2012.
31. Junheng Hao, Chuan Lei, Vasilis Efthymiou, Abdul Quamar, Fatma Ozcan, Yizhou Sun, Wei Wang, Sergiy Zhuk, and Jonathan Epperlein. Medto: Medical data to ontology matching using hybrid graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2946–2954, 2021.
32. Alberto J Cañas, Joseph D Novak, and Priit Reiska. How good is my concept map? am i a good cmapper? *Knowledge Management & E-Learning: An International Journal*, 7(1):6–19, 2015.
33. Alberto J Cañas, Priit Reiska, and Joseph D Novak. Is my concept map large enough? In *International Conference on Concept Mapping*, pages 128–143. Springer, 2016.
34. Stefan Yoshi Buhmann and Martyn Kingsbury. A standardised, holistic framework for concept-map analysis combining topological attributes and global morphologies. *Knowledge Management & E-Learning: An International Journal*, 7(1):20–35, 2015.
35. Julia Gil Llinás, Francisco Solano Macías, and Luis Manuel Tobaja Márquez. The use of concept maps as an assessment tool in physics classes: Can one use concept maps for quantitative evaluations? *Research in Science Education*, 50(5):1789–1804, 2020.
36. Robert Axelrod. *Structure of decision: The cognitive maps of political elites*, vol- ume 3276, chapter Decision for neoimperialism: the delibrations of the british eastern committee in 1918, pages 77–95. Princeton university press, 2015.
37. Robert Axelrod. *Structure of decision: The cognitive maps of political elites*, volume 3276, chapter Complexity and the presidency: gouverneur Morris in the constitutional convention, pages 96–112. Princeton university press, 2015.
38. Jac AM Vennix. *Group model building*, chapter Individual and organizational problem construction, pages 9–41. John Wiley and Sons, 1996.
39. Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. *Ad- vances in neural information processing systems*, 18, 2005.
40. Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157, 2007.
41. Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning.

*Journal of machine learning research*, 5(Aug):845–889, 2004.

1. Valerie J Shute and Diego Zapata-Rivera. Using an evidence-based approach to assess mental models. In *Understanding models for learning and instruction*, pages 23–41. Springer, 2008.
2. Alexander J Freund and Philippe J Giabbanelli. Are we modeling the evidence or our own biases? a comparison of conceptual models created from reports. In *2021 Annual Modeling and Simulation Conference (ANNSIM)*, pages 1–12. IEEE, 2021.
3. Silvia de Juan, Andres Ospina-Alvarez, Sebastián Villasante, and Ana Ruiz-Frau. A graph theory approach to assess nature’s contribution to people at a global scale. *Scientific Reports*, 11(1):1–13, 2021.
4. Lina Elsherif Ismail and Waldemar Karwowski. A graph theory-based modeling of functional brain connectivity based on eeg: A systematic review in the context of neuroergonomics. *IEEE Access*, 8:155103–155135, 2020.
5. Komal Batool and Muaz A Niazi. Towards a methodology for validation of centrality measures in complex networks. *PloS one*, 9(4):e90283, 2014.
6. Salem Alelyani, Jiliang Tang, and Huan Liu. Feature selection for clustering: A review. In Charu C. Aggarwal and Chandan K. Reddy, editors, *Data clustering: Algorithms and Applications*, pages 29–60. Chapman and Hall/CRC, 2018.
7. Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. In Charu C. Aggarwal, editor, *Data classification: Algorithms and applications*, page 37. Chapman and Hall/CRC, 2014.
8. Rizgar Zebari, Adnan Abdulazeez, Diyar Zeebaree, Dilovan Zebari, and Jwan Saeed. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, 1(2):56–70, 2020.
9. S Velliangiri and S Alagumuthukrishnan. A review of dimensionality reduction tech- niques for eﬀicient computation. *Procedia Computer Science*, 165:104–111, 2019.
10. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.
11. Saúl Solorio-Fernández, J Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2):907–948, 2020.
12. Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *2015 38th international convention on information and com- munication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.
13. Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.
14. Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
15. Manoranjan Dash, Hua Liu, and Jun Yao. Dimensionality reduction of unsupervised data. In *Proceedings ninth IEEE international conference on tools with artificial in- telligence*, pages 532–539. IEEE, 1997.
16. Roy Varshavsky, Assaf Gottlieb, Michal Linial, and David Horn. Novel unsupervised feature filtering of biological data. *Bioinformatics*, 22(14):e507–e513, 2006.
17. Monami Banerjee and Nikhil R Pal. Feature selection with svd entropy: Some modi- fication and extension. *Information Sciences*, 264:118–134, 2014.
18. Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. L2, 1-norm regularized discriminative feature selection for unsupervised. In *Twenty-Second Inter- national Joint Conference on Artificial Intelligence*, 2011.
19. Zechao Li and Jinhui Tang. Unsupervised feature selection via nonnegative spec- tral analysis and redundancy control. *IEEE Transactions on Image Processing*, 24(12):5343–5355, 2015.
20. Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, and Hanqing Lu. Unsupervised feature selection using nonnegative spectral analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 26, 2012.
21. Laurent El Ghaoui, Guan-Cheng Li, Viet-An Duong, Vu Pham, Ashok N Srivastava, and Kanishka Bhaduri. Sparse machine learning methods for understanding large text corpora. In *Proceedings of the Conference on Intelligent Data Understanding*, pages 159–173, 2011.
22. Yuanhong Li, Ming Dong, and Jing Hua. Localized feature selection for clustering.

*Pattern Recognition Letters*, 29(1):10–18, 2008.

1. Jun Guo and Wenwu Zhu. Dependence guided unsupervised feature selection. In

*Thirty-second AAAI conference on artificial intelligence*, 2018.

1. Liping Jing, Michael K Ng, and Joshua Zhexue Huang. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Transactions on knowledge and data engineering*, 19(8):1026–1041, 2007.
2. Joshua Zhexue Huang, Michael K Ng, Hongqiang Rong, and Zichen Li. Automated variable weighting in k-means type clustering. *IEEE transactions on pattern analysis and machine intelligence*, 27(5):657–668, 2005.
3. Philippe J Giabbanelli, Michael C Galgoczy, Duc M Nguyen, Romain Foy, Ketra L Rice, Nisha Nataraj, Margaret M Brown, and Christopher R Harper. Mapping the complexity of suicide by combining participatory modeling and network science. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 339–342, 2021.
4. Andrew A Tawfik, Woei Hung, and Philippe J Giabbanelli. Comparing how differ- ent inquiry-based approaches impact learning outcomes. *Interdisciplinary Journal of Problem-Based Learning*, 14(1):n1, 2020.
5. Andrew A Tawfik, Philippe J Giabbanelli, Maureen Hogan, Fortunata Msilu, Anila Gill, and Cindy S York. Effects of success v failure cases on learner-learner interaction. *Computers & Education*, 118:120–132, 2018.
6. Andrew A Tawfik. Do cases teach themselves? a comparison of case library prompts in supporting problem-solving during argumentation. *Journal of Computing in Higher Education*, 29(2):267–285, 2017.
7. Andrew Tawfik, David Jonassen, and C Keene. Why do we fall? using experiences of failure to design case libraries. *International Journal of Designs for Learning*, 3(1), 2012.
8. McKinney. pandas: a foundational python library for data analysis and statistics.

*Python for high performance and scientific computing*, 14(9):1–9, 2011.

1. Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynam- ics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
2. Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric dis- crimination: Consistency properties. *International Statistical Review/Revue Interna- tionale de Statistique*, 57(3):238–247, 1989.
3. Cindy E Hmelo-Silver. Problem-based learning: What and how do students learn?

*Educational psychology review*, 16(3):235–266, 2004.

1. John R Savery. Overview of problem-based learning: Definitions and distinctions.

*Interdisciplinary Journal of Problem-based Learning*, 1(1):9–20, 2006.

1. Stephen K Reed. The structure of ill-structured (and well-structured) problems revis- ited. *Educational Psychology Review*, 28(4):691–716, 2016.
2. Dian Permatasari Kusuma Dayu, Cerianing Putri Pratiwi, and Pratiwi Rahmah Hakim. Problem-based learning model to increase students’ critical thinking. *Jur- nal Basicedu*, 6(3):4672–4678, 2022.
3. Deniz Eseryel, Dirk Ifenthaler, and Xun Ge. Validation study of a method for assessing complex ill-structured problem solving by using causal representations. *Educational Technology Research and Development*, 61(3):443–463, 2013.
4. David H Jonassen. *Learning to solve problems: A handbook for designing problem- solving learning environments*. Routledge, 2010.
5. Gail Fitzgerald, Kevin Koury, Katherine Mitchem, Candice Hollingsead, Kevin Miller, Meeaeng Ko Park, and Hui-Hsien Tsai. Implementing case-based instruction in higher education through technology: What works best? *Journal of Technology and Teacher Education*, 17(1):31–63, 2009.
6. Hyoung Seok Shin and Allan Jeong. Modeling the relationship between students’ prior knowledge, causal reasoning processes, and quality of causal maps. *Computers & Education*, 163:104113, 2021.
7. Kun Huang, Xun Ge, and Deniz Eseryel. Metaconceptually-enhanced simulation-based inquiry: effects on eighth grade students’ conceptual change and science epistemic beliefs. *Educational Technology Research and Development*, 65(1):75–100, 2017.
8. Candace Schau, Nancy Mattern, Michael Zeilik, Kathleen W Teague, and Robert J Weber. Select-and-fill-in concept map scores as a measure of students’ connected understanding of science. *Educational and Psychological Measurement*, 61(1):136–158, 2001.
9. So Young Kwon and Lauren Cifuentes. The comparative effect of individually- constructed vs. collaboratively-constructed computer-based concept maps. *Computers & Education*, 52(2):365–375, 2009.
10. Meng Sun, Minhong Wang, Rupert Wegerif, and Jun Peng. How do students generate ideas together in scientific creativity tasks through computer-based mind mapping? *Computers & Education*, 176:104359, 2022.
11. Jihyun Si, Hyun-Hee Kong, and Sang-Hwa Lee. Developing clinical reasoning skills through argumentation with the concept map method in medical problem-based learn- ing. *Interdisciplinary Journal of Problem-Based Learning*, 13(1):5, 2019.
12. Gloria Yi-Ming Kao, Sunny SJ Lin, and Chuen-Tsai Sun. Breaking concept bound- aries to enhance creative potential: using integrated concept maps for conceptual self-awareness. *Computers & Education*, 51(4):1718–1728, 2008.
13. John C Nesbit and Olusola O Adesope. Learning with concept and knowledge maps: A meta-analysis. *Review of educational research*, 76(3):413–448, 2006.
14. Meng Yue, Meng Zhang, Chunmei Zhang, and Changde Jin. The effectiveness of con- cept mapping on development of critical thinking in nursing education: A systematic review and meta-analysis. *Nurse education today*, 52:87–94, 2017.
15. Andrew A Tawfik, Kyung Kim, and Dongho Kim. Effects of case library recommen- dation system on problem solving and knowledge structure development. *Educational Technology Research and Development*, 68(3):1329–1353, 2020.
16. Leah Frerichs, Tiffany L Young, Gaurav Dave, Doris Stith, Giselle Corbie-Smith, and Kristen Hassmiller Lich. Mind maps and network analysis to evaluate conceptualization of complex issues: A case example evaluating systems science workshops for childhood obesity prevention. *Evaluation and program planning*, 68:135–147, 2018.
17. Heisawn Jeong and Cindy E Hmelo-Silver. Seven affordances of computer-supported collaborative learning: How to support collaborative learning? how can technologies help? *Educational Psychologist*, 51(2):247–265, 2016.
18. Jan GM Kooloos, Tim Klaassen, Mayke Vereijken, Sascha Van Kuppeveld, Sanneke Bolhuis, and Marc Vorstenbosch. Collaborative group work: Effects of group size and assignment structure on learning gain, student satisfaction and perceived participation. *Medical Teacher*, 33(12):983–988, 2011.
19. H-R Pfister and Michael Oehl. The impact of goal focus, task type and group size on synchronous net-based collaborative learning discourses. *Journal of Computer Assisted Learning*, 25(2):161–176, 2009.
20. Mohammed Saqr, Jalal Nouri, and Ilkka Jormanainen. A learning analytics study of the effect of group size on social dynamics and performance in online collaborative learning. In *European Conference on Technology Enhanced Learning*, pages 466–479. Springer, 2019.
21. Allan C Jeong. Assessing change in learners’ causal understanding using sequential analysis and causal maps. In *Innovative assessment for the 21st century*, pages 187–205. Springer, 2010.
22. Allan Jeong, Haiying Li, and Andy Jiaren Pan. A sequential analysis of responses in online debates to postings of students exhibiting high versus low grammar and spelling errors. *Educational Technology Research and Development*, 65(5):1175–1194, 2017.
23. Pablo Pirnay-Dummer, Dirk Ifenthaler, and J Michael Spector. Highly integrated model assessment technology and tools. *Educational Technology Research and Devel- opment*, 58(1):3–18, 2010.
24. Philippe J Giabbanelli, Andrew A Tawfik, and Vishrant K Gupta. Learning analytics to support teachers’ assessment of problem solving: A novel application for machine learning and graph algorithms. *Utilizing learning analytics to support study success*, pages 175–199, 2019.
25. Min Kyu Kim. Models of learning progress in solving complex problems: Expertise development in teaching and learning. *Contemporary Educational Psychology*, 42:1–16, 2015.
26. Dirk Ifenthaler. Akovia: Automated knowledge visualization and assessment. *Tech- nology, knowledge and learning*, 19(1):241–248, 2014.
27. Min Kyu Kim and Kathryn S McCarthy. Using graph centrality as a global index to assess students’ mental model structure development during summary writing. *Edu- cational Technology Research and Development*, 69(2):971–1002, 2021.
28. Allan Jeong. Sequentially analyzing and modeling causal mapping processes that sup- port causal understanding and systems thinking. In *Digital Knowledge Maps in Edu- cation*, pages 239–251. Springer, 2014.
29. Karlijn L van den Broek, Sina A Klein, Joseph Luomba, and Helen Fischer. Introducing m-tool: A standardised and inclusive mental model mapping tool, 2021.
30. Rose Dieng. *Comparison of conceptual graphs for modelling knowledge of multiple experts: application to traﬀic accident analysis*. PhD thesis, INRIA, 1997.
31. Kari Chopra, Robert Rush, David Mendonça, and William A Wallace. Acquiring and assessing knowledge from multiple experts using graphical representations. In *Knowledge-Based Systems*, pages 293–326. Elsevier, 2000.
32. Jill Fielding and Katie Makar. Challenging conceptual understanding in a complex sys- tem: supporting young students to address extended mathematical inquiry problems. *Instructional Science*, pages 1–27, 2021.
33. Arthur Bakker. Discovery learning: zombie, phoenix, or elephant? *Instructional Science*, 46(1):169–183, 2018.
34. Gary Chartrand and Ping Zhang. *A first course in graph theory*. Courier Corporation, 2013.
35. Vincenzo Carletti. *Exact and Inexact Methods for Graph Similarity in Structural Pattern Recognition*. PhD thesis, Université de Caen; Universita degli studi di Salerno, 2016.
36. Ömer Nebil Yaveroğlu, Tijana Milenković, and Nataša Pržulj. Proper evaluation of alignment-free network comparison methods. *Bioinformatics*, 31(16):2697–2704, 2015.
37. Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. Comparing meth- ods for comparing networks. *Scientific reports*, 9(1):1–19, 2019.
38. Payam Aminpour, Heike Schwermer, and Steven Gray. Do social identity and cogni- tive diversity correlate in environmental stakeholders? a novel approach to measuring cognitive distance within and between groups. *Plos one*, 16(11):e0244907, 2021.
39. Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance.

*Pattern Analysis and applications*, 13(1):113–129, 2010.

1. Pablo Pirnay-Dummer. *Expertise und Modellbildung-MITOCAR*. PhD thesis, Freiburg (Breisgau), Univ., Diss., 2006, 2006.
2. Bertram Gawronski. Back to the future of dissonance theory: Cognitive consistency as a core motive. *Social cognition*, 30(6):652–668, 2012.
3. Shinobu Kitayama, Hannah Faye Chua, Steven Tompson, and Shihui Han. Neural mechanisms of dissonance: An fmri investigation of choice justification. *Neuroimage*, 69:206–212, 2013.
4. Michael L Epstein, Amber D Lazarus, Tammy B Calvano, Kelly A Matthews, Rachel A Hendel, Beth B Epstein, and Gary M Brosvic. Immediate feedback assessment tech- nique promotes learning and corrects inaccurate first responses. *The Psychological Record*, 52(2):187–201, 2002.
5. Michael L Epstein and Gary M Brosvic. Students prefer the immediate feedback assessment technique. *Psychological reports*, 90(3\_suppl):1136–1138, 2002.
6. Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
7. Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245, 2014.
8. J Michael Spector. Cognition and learning in the digital age: Promising research and practice. *Computers in human behavior*, 24(2):249–262, 2008.
9. Arthur C Graesser, Murray Singer, and Tom Trabasso. Constructing inferences during narrative text comprehension. *Psychological review*, 101(3):371, 1994.
10. Sidney D’Mello, Blair Lehman, Reinhard Pekrun, and Art Graesser. Confusion can be beneficial for learning. *Learning and Instruction*, 29:153–170, 2014.
11. Blair Lehman, Sidney D’Mello, Amber Strain, Caitlin Mills, Melissa Gross, Allyson Dobbins, Patricia Wallace, Keith Millis, and Art Graesser. Inducing and tracking con- fusion with contradictions during complex learning. *International Journal of Artificial Intelligence in Education*, 22(1-2):85–105, 2013.
12. Laurent S Daloz. *Effective teaching and mentoring*. Number 374.973 D144e. California, US: Bass Publishers, 1986.
13. Kaspar Riesen. Structural pattern recognition with graph edit distance. *Advances in computer vision and pattern recognition. Springer, Cham*, 2015.
14. Woei Hung, David H Jonassen, Rude Liu, et al. Problem-based learning. *Handbook of research on educational communications and technology*, 3(1):485–506, 2008.
15. Joseph Klein. The failure of a decision support system: inconsistency in test grading by teachers. *Teaching and Teacher Education*, 18(8):1023–1033, 2002.
16. Peggy A Ertmer, Krista D Glazewski, Donna Jones, Anne Ottenbreit-Leftwich, Yuk- sel Goktas, Kelly Collins, and Aslihan Kocaman. Facilitating technology-enhanced problem-based learning (pbl) in the middle school classroom: An examination of how and why teachers adapt. *Journal of Interactive Learning Research*, 20(1):35–54, 2009.
17. National Research Council et al. Developing assessments for the next generation science standards. 2014.
18. Michael D McNeese and Phillip J Ayoub. Concept mapping in the analysis and design of cognitive systems: a historical review. *Applied concept mapping: Capturing, analyzing, and organizing knowledge*, 47, 2011.
19. Alla Anohina-Naumeca. *Concept Map-Based Formative Assessment of Students’ Struc- tural Knowledge: Theory and Practice*. Cambridge Scholars Publishing, 2019.
20. Ha Nguyen and Rossella Santagata. Impact of computer modeling on learning and teaching systems thinking. *Journal of Research in Science Teaching*, 58(5):661–688, 2021.
21. Kirsten E de Ries, Harmen Schaap, Anne-Marieke MJAP van Loon, Marijke MH Kral, and Paulien C Meijer. A literature review of open-ended concept maps as a research instrument to study knowledge and learning. *Quality & Quantity*, pages 1–35, 2021.
22. Björn Rohles. *From Tool to Experience: Establishing a User Experience Perspective on Digital Concept Mapping*. PhD thesis, University of Luxembourg, Luxembourg, 2021.
23. Bill Cope, Mary Kalantzis, and Duane Searsmith. Artificial intelligence for education: Knowledge and its assessment in ai-enabled learning ecologies. *Educational Philosophy and Theory*, 53(12):1229–1245, 2021.
24. Katja Weinerth, Vincent Koenig, Martin Brunner, and Romain Martin. Concept maps: A useful and usable tool for computer-based knowledge assessment? a literature review with a focus on usability. *Computers & Education*, 78:201–209, 2014.
25. Steffen Wagner, Karel Kok, and Burkhard Priemer. Measuring characteristics of ex- planations with element maps. *Education Sciences*, 10(2):36, 2020.
26. Saad T Hasson and Zahraa Hussein. Correlation among network centrality metrics in complex networks. In *2020 6th International Engineering Conference “Sustainable Technology and Development”(IEC)*, pages 54–58. IEEE, 2020.
27. Sofya Kogotkova, Milena Oehlers, Tatiana Ermakova, and Benjamin Fabian. Correla- tion analysis of local graph metrics. *Available at SSRN 3192028*, 2018.
28. Alexander J Freund and Philippe J Giabbanelli. An experimental study on the scal- ability of recent node centrality metrics in sparse complex networks. *Frontiers in big Data*, 5, 2022.
29. Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, pages 1–31, 2022.
30. Elizabeth Sturgiss, Thea Luig, Denise L Campbell-Scherer, Richard Lewanczuk, and Lee A Green. Using concept maps to compare obesity knowledge between policy makers and primary care researchers in canada. *BMC research notes*, 12(1):1–6, 2019.
31. Paul W Hodges, Jacek Cholewicki, John M Popovich Jr, Angela S Lee, Payam Amin- pour, Steven A Gray, Michael T Cibulka, Mel Cusi, Brian F Degenhardt, Gary Fryer, et al. Building a collaborative model of sacroiliac joint dysfunction and pelvic girdle pain to understand the diverse perspectives of experts. *PM&R*, 11:S11–S23, 2019.
32. Karen Kelm and Francois V Bolduc. How knowledge mapping can bridge the commu- nication gap between caregivers and health professionals supporting individuals with complex medical needs: A study in fragile x syndrome. *Frontiers in Psychiatry*, 12, 2021.
33. Brian G Giles, C Scott Findlay, George Haas, Brenda LaFrance, Wesley Laughing, and Sakakohe Pembleton. Integrating conventional science and aboriginal perspectives on diabetes using fuzzy cognitive maps. *Social science & medicine*, 64(3):562–576, 2007.
34. Pablo Pirnay-Dummer, Dirk Ifenthaler, and Norbert M Seel. Designing model-based learning environments to support mental models for learning. *Theoretical foundations of learning environments*, 2:66–94, 2012.
35. Samvel Mkhitaryan and Philippe J Giabbanelli. How modeling methods for fuzzy cognitive mapping can benefit from psychology research. In *2021 Winter Simulation Conference (WSC)*, pages 1–12. IEEE, 2021.
36. Nina Bonderup Dohn and Lina Markauskaite. Conceptions of transfer, transformation and resituation. *Designing for situated knowledge transformation*, pages 39–58, 2019.
37. Carla Limongelli, Alessandro Marani, Filippo Sciarrone, and Marco Temperini. Mea- suring the similarity of concept maps according to pedagogical criteria. *IEEE Access*, 10:27655–27669, 2022.
38. Omar Elsayed, Carla Limongelli, Filippo Sciarrone, Matteo Lombardi, Alessandro Marani, and Marco Temperini. An on-line framework for experimenting with con- cept maps. In *2019 18th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–8. IEEE, 2019.
39. Sharon Dole, Lisa Bloom, and Kristy K Doss. Engaged learning: Impact of pbl and pjbl with elementary and middle grade students. *Interdisciplinary Journal of Problem- Based Learning*, 11(2):9, 2017.
40. Steven Netcoh and Penny A Bishop. Personalized learning in the middle grades: A case study of one team’s successes and challenges. *Middle Grades Research Journal*, 11(2), 2017.
41. David H Jonassen. Instructional design models for well-structured and iii-structured problem-solving learning outcomes. *Educational technology research and development*, 45(1):65–94, 1997.
42. Jeroen JG Van Merriënboer, Richard E Clark, and Marcel De Croock. Blueprints for complex learning: The 4c/id-model. *Educational technology research and development*, 50(2):39–61, 2002.
43. Elena Novak, Jerry Daday, and Kerrie McDaniel. Assessing intrinsic and extraneous cognitive complexity of e-textbook learning. *Interacting with Computers*, 30(2):150– 161, 2018.
44. Andrew A Tawfik, Jessica Gatewood, Jaclyn J Gish-Lieberman, and Andrew J Hamp- ton. Toward a definition of learning experience design. *Technology, Knowledge and Learning*, pages 1–26, 2021.
45. Matthew Schmidt, Yvonne Earnshaw, Andrew A Tawfik, and Isa Jahnke. Methods of user centered design and evaluation for learning designers. *Learner and user experience research*, 2020.
46. John Ruppert, Ravit Golan Duncan, and Clark A Chinn. Disentangling the role of domain-specific knowledge in student modeling. *Research in Science Education*, 49(3):921–948, 2019.