

## **ABSTRACT**

### **EFFICIENTLY COMPARING MENTAL REPRESENTATIONS: VISUALIZING AND MATCHING CAUSAL NETWORKS**

Vishrant Krishna Gupta, M.S.  
Department of Computer Science  
Northern Illinois University, 2018  
Dr. Reva Freedman, Director

A concept map can be used to represent knowledge. A causal map is a type of concept map and a graphical way that can be used to define thinking of an individual about a system by using links between nodes in a network to represent causation or influence. A causal map thus can be used to solve ill-structured problems and evaluate student's assignment. Causal maps can also be used to generate feedback for a student by comparing it with an expert map. However, exact network comparison is an NP-complete problem and is known to be graph isomorphism problem.

In order to solve the problem, an online platform is created, where instead of exact graph comparison mathematical techniques such as graph edit distance, graph kernel, and graph embedding are used to measure how closely two networks match. Before performing comparison it is important to establish a common set of terms between two networks, as a student term might have a slight variations in the way they name the included concepts in causal map. This requires alignment of terms which is often contextual, the process is eased by implementing a recommendation system by leveraging alignments done by other users. Step by step feedback is generated to guide the student in developing their understanding of a system. This software can be used in Massive Open Online Course (MOOC) and classroom-based system, where assessing the learning experience and providing feedback to a large number of participants is an ongoing challenge.

NORTHERN ILLINOIS UNIVERSITY  
DE KALB, ILLINOIS

AUGUST 2018

**EFFICIENTLY COMPARING MENTAL REPRESENTATIONS: VISUALIZING AND  
MATCHING CAUSAL NETWORKS**

BY

VISHRANT KRISHNA GUPTA  
© 2018 Vishrant Krishna Gupta

A THESIS SUBMITTED TO THE GRADUATE SCHOOL  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE  
MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

Thesis Director:  
Dr. Reva Freedman

## **ACKNOWLEDGEMENTS**

This thesis became a reality with the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

First, I would like to express my sincere gratitude to my advisor, Dr. Reva Freedman. I thank her not only for the continuous support of my Master's study and related research, but also for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my master's study.

Besides my advisor, I would like to thank the rest of my thesis committee, Dr. Hamed Alhoori and Dr. Nicholas Karonis, for their insightful comments, encouragement, and their questions which encouraged me to widen my research from a different perspective. I would like to extend my thanks to Dr. Philippe Giabbanelli for his support.

Thanks to our collaborator, Dr. Andrew Tawfik, for providing the data to analyze the students' mental model with experts'.

Finally, I am grateful to my team members Russell Lankenau, David Offord, and Adam Wykle for helping me to implement a module on improving student's maps. I thank Magda Baniukiewicz, Mrinal Kanti Roy, Eric Lavin, and Nick Rosso, for the stimulating discussions and for all the fun we have had during the course of my master's.

## **DEDICATION**

To my parents, for always supporting and believing on me.

*“Would you like me to give you a formula for success? It’s quite simple, really: Double your rate of failure. You are thinking of failure as the enemy of success. But it isn’t at all. You can be discouraged by failure or you can learn from it, so go ahead and make mistakes. Make all you can. Because remember that’s where you will find success.”*

- THOMAS J. WATSON, JR.,  
*Speech at the University of Pennsylvania,*  
*1973*

# CONTENTS

	Page
List of Tables . . . . .	viii
List of Figures . . . . .	ix
Chapter	
1 Introduction . . . . .	1
1.1 Contributions . . . . .	4
1.2 Outline . . . . .	5
1.2.1 Leverage HCI to align terms efficiently and informatively . . . . .	5
1.2.2 Develop a collaborative approach to aligning terms . . . . .	7
1.2.3 Perform comprehensive network comparisons . . . . .	7
2 Background: Advantages and challenges of software solutions for the structural assessment of knowledge . . . . .	12
3 An online environment to compare students' and expert solutions to ill-structured problems . . . . .	17
3.1 Introduction . . . . .	18
3.2 Background . . . . .	21
3.2.1 Computational approaches to assessment . . . . .	21
3.3 Software functionalities and architecture . . . . .	24
3.3.1 Overview . . . . .	24
3.3.2 Architecture and technologies . . . . .	27
3.3.3 MySQL database . . . . .	32

Chapter	Page
3.4 Alignment process . . . . .	33
3.4.1 Visualization. . . . .	33
3.4.2 Aggregate recommendation . . . . .	34
3.5 Discussion . . . . .	37
4 Digital tools to guide students in answering ill-structured problems . . . . .	42
4.1 Introduction. . . . .	43
4.2 Background. . . . .	46
4.2.1 Developing knowledge structures through ill-structured problem solving .	46
4.2.2 Assessment in ill-structured problem solving. . . . .	48
4.3 Using graph comparison methods for assessment . . . . .	50
4.3.1 An intuitive introduction to graph comparison . . . . .	50
4.3.2 Graph Edit Distance (GED) . . . . .	51
4.3.3 Graph Kernels. . . . .	54
4.3.4 Graph Embeddings. . . . .	56
4.4 Implementation . . . . .	57
4.5 Discussion . . . . .	59
5 Digital tool to improve student’s map by providing them with feedback . . . . .	63
5.1 Introduction. . . . .	63
5.2 Disjoint paths and loops. . . . .	65
5.2.1 Implementation of disjoint path and loops. . . . .	68
5.3 Design Process. . . . .	69
6 Conclusion . . . . .	73
6.1 Contribution . . . . .	74
6.2 Limitations . . . . .	76

Chapter	Page
6.3 Future work. . . . .	77
6.3.1 Collaborative filtering based recommendation system . . . . .	77
6.3.2 Clustering students. . . . .	77
Bibliography . . . . .	79



## LIST OF TABLES

Table		Page
3.1	Client side architecture . . . . .	31
3.2	Server side architecture . . . . .	32

## LIST OF FIGURES

Figure	Page
1.1 Identical terms in two maps. . . . .	2
1.2 Previous software by Giabbanelli & Tawfik. . . . .	3
1.3 Proposed interactive visualization. . . . .	6
1.4 Possible edit path. . . . .	9
1.5 Graph with cycle. . . . .	10
1.6 Graph embedding. . . . .	11
3.1 Signing up to create an account. . . . .	25
3.2 Creation of an assignment. . . . .	26
3.3 Alignment of terms. . . . .	28
3.4 Visualization of aligned factors. . . . .	29
3.5 Visualization using D3Plus. . . . .	30
3.6 Model, View, Controller (MVC) based N-tier system architecture. . . . .	30
3.7 Diagram of the database. . . . .	40
3.8 Main steps of our algorithm to perform an aggregate recommendation. . . . .	41
3.9 Network built in Neo4J for aggregate recommendation. . . . .	41
4.1 Concept map. . . . .	45
4.2 Sequences of operations to transform maps. . . . .	53
4.3 Comparison of two maps based on cycle distribution. . . . .	55
4.4 Newest version of the ITACM software. . . . .	60

Figure	Page
5.1 Example: Disjoint path. . . . .	66
5.2 Example: A balancing loop. . . . .	67
5.3 Wireframe of the proposed solution to provide feedback. . . . .	71
5.4 An example of how feedback is provided. . . . .	72
6.1 Demonstration of a factor influencing another factor. . . . .	74
6.2 Variation in terms of student and an expert in causal map. . . . .	75
6.3 The possible number of comparison between a term of student's and expert map. .	75
6.4 Clustering students based on different graph matrices. . . . .	78

# CHAPTER 1

## INTRODUCTION

Problems in the classroom are often well-structured as they have predefined answers. Assessing a student's work thus consists of contrasting the student's answer to the set of valid answers. However, the problems faced by practitioners are frequently ill-structured problems (ISP) as they admit a much broader set of solutions, which are often contextual [67], i.e., they might depend on the context to which a solution is applied that includes preferences and values. To solve an ISP, students use skills in hypothesis generation and causal reasoning. That is, they form a mental representation that integrates interacting and competing context-dependent factors, and then evaluate the consequences of possible actions based on this mental representation. Since instructors cannot assess the mental representation internally held by students, it is typically externalized in the form of dialogues or essays. However, instructors lack scalable and reliable technological solutions to assess such forms of solutions, and this assessment challenge limits the use of ISPs in the classroom [111], making it seem supplemental to coursework instead of being a driver of instruction [84].

A mental representation of a problem can also be externalized in the form of a causal network, which articulates causal factors as well as their antecedents and consequents Figure 1.1. Many software tools allow students to generate causal networks (e.g., Coggle). Tools are also starting to emerge to help with their assessment [32, 56]. Given the open-ended and context-sensitive nature of ISPs, one way to assess causal maps is by contrasting a student's map with an expert's map. Studies on problem solving have revealed that expert and novice understanding differ in several ways, including a greater understanding of causality and conditional rules [48], as well as foreseeing multiple causal processes [58]. While this provides a clear rationale to contrast student's and

expert causal networks, a perfect comparison is computationally infeasible. For instance, finding the largest part of a student’s network that matches the expert’s (i.e., sub-graph isomorphism) is an NP-complete problem [8]. The comparison becomes feasible when the networks have unique concept names [16], but terms in the students’ networks are unlikely to be exactly the same as the expert’s (Figure 1.1). Giabbanelli and Tawfik recently developed software to align the student’s terms with the expert’s terms (Figure 1.2) [32]. However, the software requires extensive work by humans to perform alignments for every term as illustrated in Figure 1.2c, and it provides only rudimentary metrics after the alignment is performed (e.g., number of nodes or edges in common).

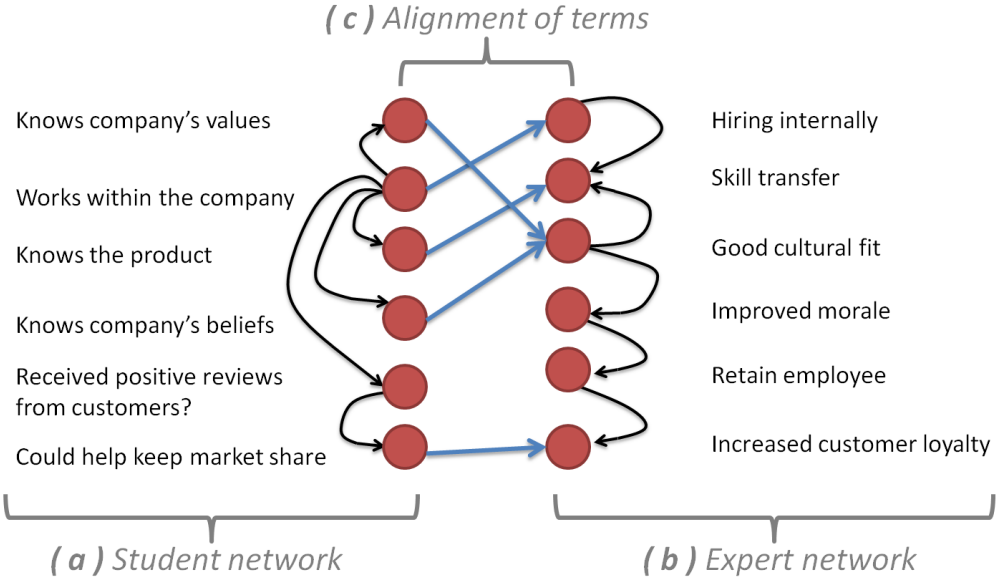


Figure 1.1: Terms in two maps may not be identical due to variations in language. The alignment problem consists of mapping the terms of students onto expert terms, thus allowing for network comparison.

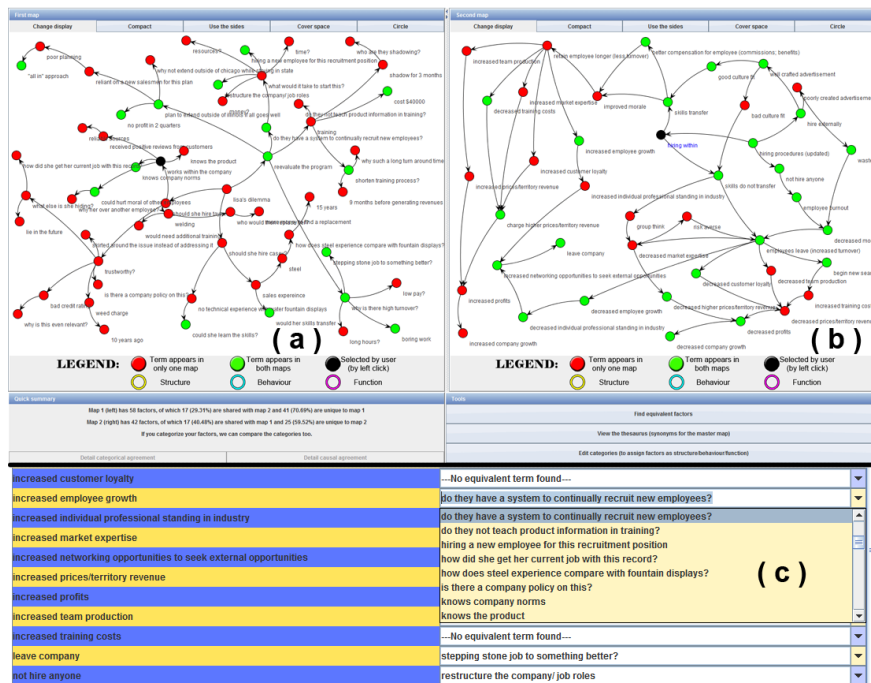


Figure 1.2: The software by Giabbanelli and Tawfik allows one to visualize one student network (a) and the expert network (b). The alignment is performed by manually selecting the corresponding expert term in a drop-down menu (c).

## 1.1 Contributions

The focus of this thesis is to efficiently compare mental representations in the form of causal networks, with applications to the assessment of ill-structured problem (ISP) solving in an educational context. This goal will be accomplished through the completion of three specific aims:

1. Leverage human-computer interaction (HCI) to align expert and students' terms efficiently (in terms of human efforts) and informatively (in terms of the insight for learning generated through the alignment process).
2. Develop a collaborative approach to aligning terms through an online platform and the use of information retrieval techniques.
3. Perform comprehensive network comparisons, going beyond simple metrics.

By accomplishing these aims, we will make it possible for educators to assess mental representations in a systematic and scalable manner. The short-term benefits will be the release of online open-access software with which educators can upload and compare students' maps with expert maps. In the long term (i.e., beyond the scope of the present work), this thesis will contribute to the application of computational methods in the difficult problem of comparing mental representations, which is found in many fields beyond only educational assessment. For instance, identifying the extent to which individuals think 'similarly' is essential for setting up groups in randomized controlled trials (RCTs), where the distribution of perspectives should be similar at baseline across groups [30]. Similarly, knowing the 'distance' between mental representations is important for contrasting the perspectives of stakeholders in problems requiring participative decision-making, such as ecological management [36, 41].

## 1.2 Outline

This thesis combines several fields: human-computer interaction (HCI), information retrieval (IR), and network science. The specific techniques within each field are briefly described in the context of the specific aims for which they are used. Network science and studies on mental representation constitute the common thread which guides this work, given that we approach mental representations in the form of networks (Figure 1.1) rather than in narrative forms (e.g., essays) or through live discussions. The first aim uses HCI to improve the alignment of terms (i.e., mapping between network nodes). Separately, the second aim uses IR, and specifically recommender systems, to suggest the best alignments. Upon completion of the first two aims, the last one then performs comprehensive network comparisons.

### 1.2.1 Leverage HCI to align terms efficiently and informatively

Our aim is to design interactive visualizations that allow humans to efficiently and informatively align terms. This translates to two specific criteria. First, usability should support each contributor in efficiently aligning terms, through a minimal number of clicks and a low cognitive load. Second, the alignment itself should be informative: the structure of the alignment as shown in Figure 1.1 can hold insightful information, for instance when seeing a one-to-one mapping compared to a many-to-one (i.e., when several student terms map onto the same expert term).

Interactive visualizations can help navigate datasets with many combinations [29], which means:

In many disciplines, data and model scenarios are becoming multifaceted: data are often spatiotemporal and multivariate; they stem from different data sources (multimodal data), from multiple simulation runs (multirun/ensemble data), or from multiphysics sim-



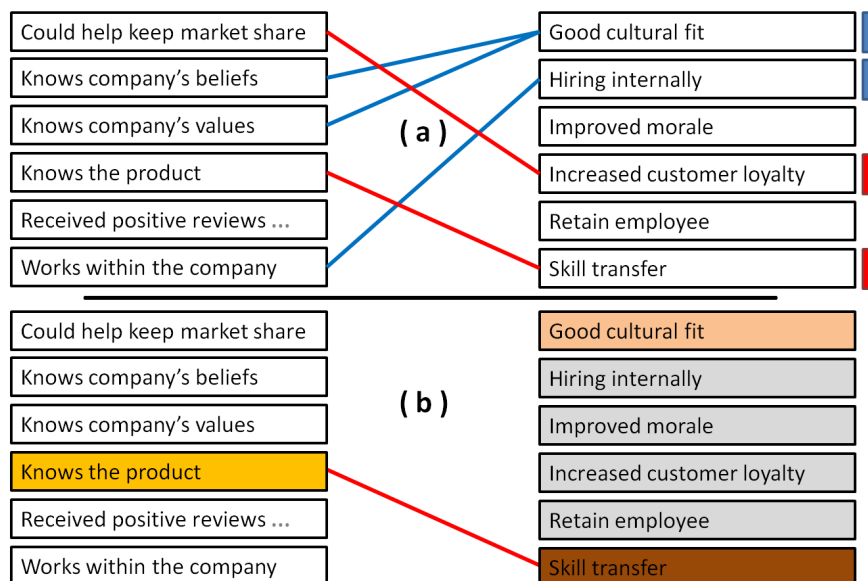


Figure 1.3: Proposed interactive visualization showing all alignments (a), and a user clicking on a specific student term (b).

ulations of interacting phenomena (multimodel data resulting from coupled simulation models). [74]

But even readily existing solutions are underutilized when it comes to text [28]. Here, we propose to treat the expert's and students' terms as two dimensions and use interactive multi-tiered visualizations (also used by Starlight [101], Jigsaw [119], and Eduvis [70]) to see and edit matches. The structure of the matches becomes apparent (Figure 1.3a vs. Figure 1.2c), and also shows whether matches are confirmed by users (shown in blue connections) or suggested (shown in red connections). Suggestions leverage the online collaborative environment to highlight whether matches have occurred in other assignments (frequency shown by hue; Figure 1.3b) or not (greyed out). A match requires at least two clicks to set the source and endpoint. Our proposed visualization meets this minimum, as users only click on students' terms and then the corresponding expert's term to set a match.

In sum, we will use interactive multi-tiered visualizations to (i) minimize the number of clicks to perform alignments, and (ii) offer insight about the alignment as it progresses.

## **1.2.2 Develop a collaborative approach to aligning terms**

Giabbanelli and Tawfik previously pointed out that an alignment (Figure 1.2) can be used not only to make two networks comparable, but also to speed up future alignment processes [32]. The intuition is that if two terms have been found equivalent previously, then the computer may already align them and simply flag this alignment as requiring confirmation (Figure 1.3; red links) rather than asking users to do it all again. In other words,

The more educators assess causal maps, and the more it generates terminologies as by-product of the analysis; therefore, future analyses become increasingly automatized. [32]

However, previous equivalences may have been established in different contexts (e.g., “formal” and “mathematical notation” may be equivalent in one context, but “formal” and “fancy reception” may be equivalent in another), and multiple equivalences may exist. The computer thus has to identify the best candidate, that is, recommend the most likely match.

We will thus use the technique of recommender systems, which is widely used in contexts such as e-commerce where recommended purchases leverage the patterns of previous customers’ purchases [59]. Recommender systems fall into two broad categories [95]: collaborative filtering (focusing on similarities between users) and content-based (focusing on properties of the items). Most modern recommender systems are hybrids, combining these ideas. We will thus use a hybrid system.

## **1.2.3 Perform comprehensive network comparisons**

A graph  $g$  is defined as set of vertices ( $V$ ) and edges ( $E$ ) and vertices and edge labeling functions  $u$  and  $v$  respectively. Graph comparison is a difficult problem for which there exist two broad

categories of approaches: exact graph matching, and inexact graph matching (also referred as error-tolerant graph matching) [129].

Exact graph matching algorithms aim to find whether two graphs are partially or entirely identical in terms of structure and labels. Being identical means that there exists a bijective function called an isomorphism [24] that maps one graph to the other while preserving adjacencies (e.g., if two nodes shared a link in graph A, then their corresponding nodes in graph B should also share a link). Formally, two graphs are said to be isomorphic if there exists an isomorphism between them and it is denoted by  $g1 \cong g2$ . Riesen highlighted the difficulty of isomorphisms:

Subgraph isomorphism is an NP-complete problem and is a harder problem than graph isomorphism, as we have to not only check whether a permutation of  $g1$  is identical to  $g2$ , but also decide whether  $g1$  is isomorphic to any of the subgraphs of  $g2$  with equal size as  $g1$ . [96]

Inexact graph matching algorithms aim at finding partial isomorphisms. Several algorithms have been proposed in this category. A simple approach could be to use a depth-first search, which traverses a graph starting at a root node. If traversals in two graphs yield similar sequences of nodes, then we could conclude that the graphs might be similar. The space complexity of depth-first search is  $O(|V|)$  in the worst case and time complexity of  $O(|V| + |E|)$  [13] when implemented using an adjacency list. While this solution is computationally cheap, it produces false positives (two traversals may be identical but the underlying graphs aren't) as well as false negatives (two traversals may be different not because the graphs are different but because they use different starting points).

Inexact graph matching algorithms can be applied to scenarios where it is possible that labels for vertices or edges differ but are still considered equivalent, or that there are some extra or missing vertices or edges but the graph structures are not substantially different. That is, inexact graph matching can be applied to more problems than exact graph matching (which is a special

case of inexact matching). In this thesis, we will use inexact graph matching techniques to compare the causal maps of students and experts.

Intuitively, one approach to inexact graph matching is to transform the source graph into the target graph in steps, where errors at each step incur a ‘penalty’, and the total cost of the transformation consists of the cumulative penalties incurred (Figure 1.4). A straightforward transformation would be to delete all nodes of the source graph, then add all nodes of the target graphs, and then the edges. That is, a graph can always be transformed into another. Thus, the objective is not to find just any transformation, but to find one with low cost, although not absolutely minimal (as it then would become an exact graph matching). One method using this approach is to use the Graph Edit Distance (GED) [97], which itself can be calculated using a variety of algorithms [100]. We will review and contrast options, and then select one approach to using the GED in this thesis.

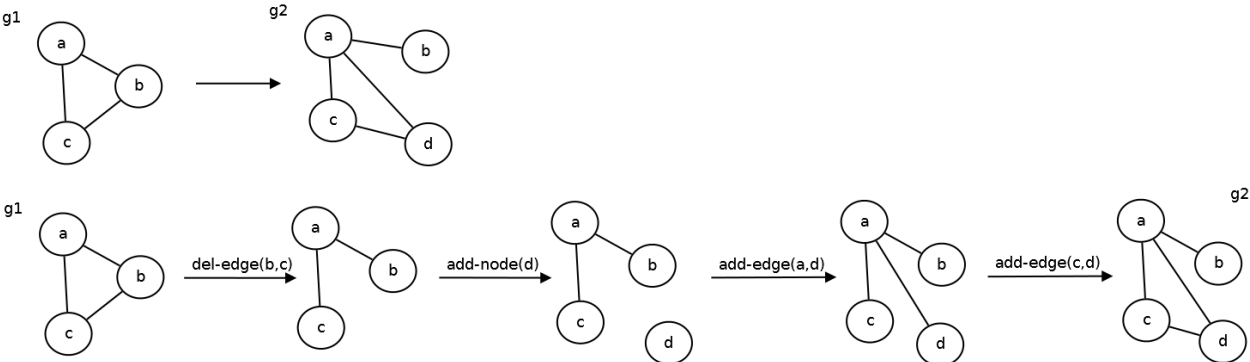


Figure 1.4: A possible edit path for transforming graph g1 to g2. Graph g1 is transformed to g2.

I will try other approaches like graph kernels and graph embeddings. An illustration for graph kernels is shown in Figure 1.5. It is a similarity function between the given graphs. The function is computable in polynomial time. Several methods exist to create graph kernels, as it is up to the user to identify what relevant ‘features’ (or ‘kernels’) need to be extracted. A graph kernel between two given graphs can be found with a random walk kernel, which conceptually performs random walks on the given graphs simultaneously, then counts the number of paths produced by both walks. This is equivalent to doing random walks on the direct product of the pair of graphs,

and thus a kernel can be derived that can be efficiently computed [130]. Another approach can be to get the number of cyclic patterns in the structure: for instance, a molecular structure can be represented as cycles ( $C1 \neq C2$ ) and this structure can be compared with other structures only in terms of cycles (Figure 1.5). Other approaches like graph kernels and graph embeddings can be used to contrast two given graphs. An illustration for the graph kernel is shown in Figure 1.5. It is a similarity function between given graphs. The function is computable in polynomial time.

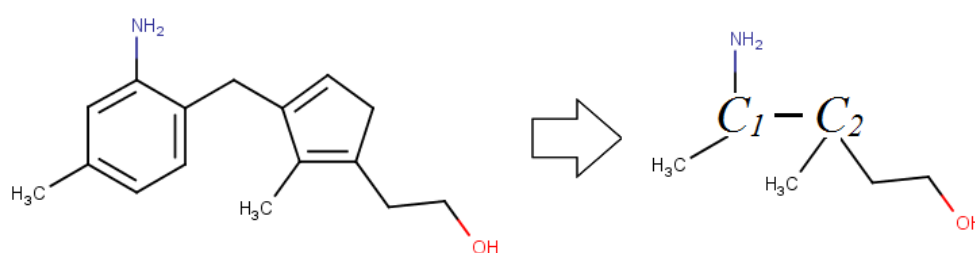


Figure 1.5: Cyclic molecular graph and its relevant cycle hypergraph.

In graph embedding we represent a graph based on its features, for example, number of nodes ( $n$ ), number of edges ( $v$ ) or average degree, as a point in a suitable vector space, as illustrated in Figure 1.6. Interesting embeddings are characterized by the fact that they preserve the similarity of the graphs: the more two graphs are similar, the closer the corresponding points are in the target vector space. Although seminal works in this area were already present in earlier literature, it is in the last decade that these techniques have gained popularity in the pattern recognition community.

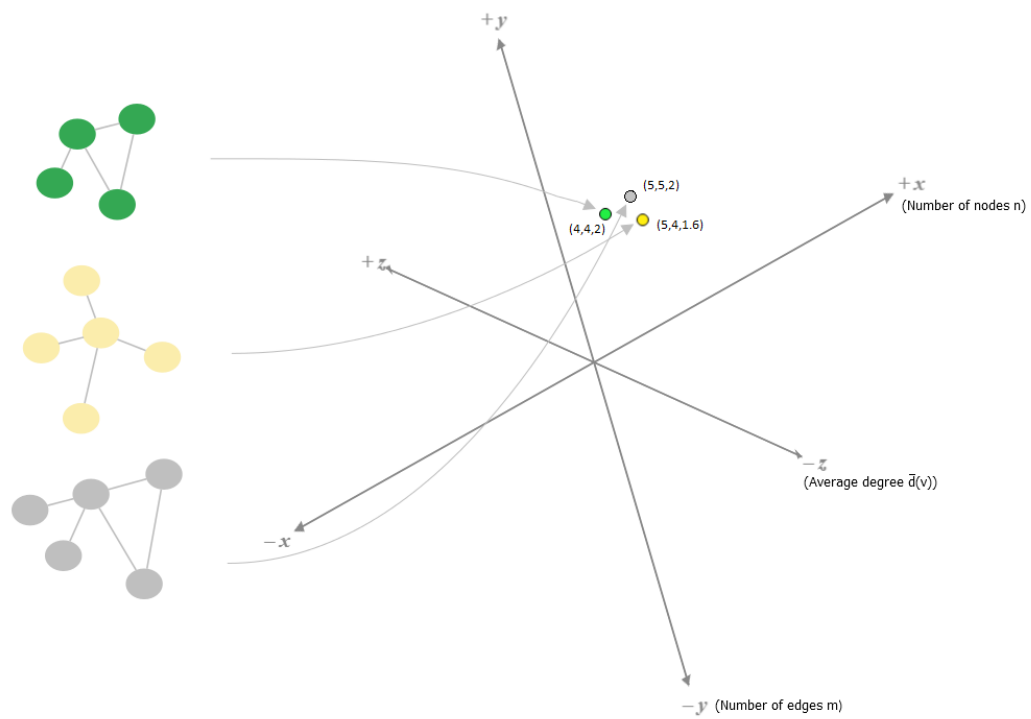


Figure 1.6: Embedding of a graph using the number of nodes, number of edges, and average degree.

## **CHAPTER 2**

### **BACKGROUND: ADVANTAGES AND CHALLENGES OF SOFTWARE SOLUTIONS FOR THE STRUCTURAL ASSESSMENT OF KNOWLEDGE**

The field of research concerned with examining and representing the way learners organize their knowledge of a subject is called structural assessment of knowledge (SAK) [128]. When new information is acquired, it is incorporated into the learner's existing body of knowledge, and associations with existing knowledge are formed [128]. These associations may be correct, incorrect, or partially correct at any given time during the learning process. SAK aims to provide a procedure for representing the concepts and relationships surrounding a topic of interest using various methods of knowledge representation, as well as methods for evaluating the correctness of those representations [128]. The specific branch of SAK that we focus on in this paper is based on cognitive maps.

Cognitive maps (also called concept maps) consist of a set of concepts, as well as a set of relationships linking these concepts. They are typically represented using node and link diagrams, with nodes representing individual concepts, and links representing the relationship connecting two concepts [89]. Specialized variations of cognitive maps have been created for many different purposes, typically by constraining the types of relationships allowed between nodes, or by imposing a more rigid hierarchical structure on the map [128]. These constraints can simplify the process of evaluating or comparing maps [87], provide a more structured process for creating new maps [128], or limit the scope of the mapping activity for pedagogical reasons [49].

The most general form of cognitive map consists of a set of concepts connected with lines representing specific relationships identified by the individual creating the map. This type of map was initially developed by Novak and Gowin in the 1980s [89], and provides the learner with the most flexibility in representing their own perception of the material, since no fixed structure is dictated, and no restriction is placed on the available relationships [49]. While useful as a learning tool, these maps can be difficult to evaluate, so additional constraints are imposed to simplify the resulting maps for the instructors or the software evaluating them [87].

Knowledge maps and mind maps restrict the allowed structures in two different ways, with knowledge maps restricting the set of allowed relationships, and mind maps constraining the overall structure into a hierarchical format centered on a single concept. Pathfinder networks are created by ranking the degree of relationship between each pair of concepts, and then applying a weighting algorithm to determine which of these relationships should be included [128]. Finally, causal maps require that all relationships be directed (i.e., from one concept to another, rather than bidirectional), and limit the allowed relationships to only directed positive and negative causation, with positive causation indicating that an increase in the strength of one concept leads to an increase in the strength of the related concept, and negative causation indicating that an increase in one concept leads to a decrease in the other.

With any of the types of structured mentioned above, additional restrictions can be placed on the form of the maps created by the instructors using them to simplify the work to be performed by either the students or the instructor [128]. Ho et al. suggest that having students fill in a partially constructed concept map can reduce the cognitive load of the creation process, allowing students to focus on the concepts being learned rather than the process being used [49]. Various studies state that restricting the available concepts in a given map can make them easier to evaluate [49, 87, 134].

One popular use of concept maps in SAK is as an assessment tool. Trumppower and Vanapalli refer to this as “SAK of Learning” in contrast to the other types of SAK which will be described later. In this context, concept maps would be created or completed in place of a more traditional



exam or essay at the end of an instructional topic to evaluate how well students had mastered the material [128]. Because concept maps are being used in place of a more traditional assessment instrument, it is critically important that the method of evaluating maps is reliable and accurately indicates the degree of mastery attained [132]. This can be especially difficult because concept maps by their very nature represent an individual view of a particular subject area [87]. Instructors can also be resistant to the use of concept maps as assessment tools because they perceive them to be time-consuming to evaluate, and students because they are time-consuming to produce [49]. Whether this is true in practice is not certain. McClure, Sonak, and Suen did a study on evaluation of concept maps by hand, using six different methods, both with and without a referent map for comparison, and found that none of the methods required more than 5 minutes per map on average, which they judged to be similar to the time required to evaluate an essay, based on personal experience [87].

To be truly usable as an assessment tool, it would be preferable to have a reliable, automated method of evaluating concept maps [134]. Most previous research has focused on automatic evaluation of constrained concept maps. The Pathfinder software generates both configural and common similarity scores to show how closely a student map matches a referent, but this is, of course, limited only to Pathfinder networks [128]. Both of these metrics show how closely the relationships defined by the student match the relationships present in a referent map. Wu et al. performed a study evaluating the use of automatic scoring and feedback generation using CMapTools, a popular concept mapping application. In this case, it was the set of terms that was constrained, so that meaningful comparisons could be made between maps [134]. Another common technique used in the evaluation of concept maps is the use of rubrics defined for each concept mapping task. Using rubrics has the advantage that different, context-dependent features can be identified as important in specific assessments. Rubric-based evaluation is, however, primarily a manual task, since the judgement of whether a student has satisfied each item is highly subjective [87, 128].

Concept maps can also be used as a pedagogical tool rather than an assessment tool. Trumpower and Vanapalli make a distinction between “SAK as Learning” and “SAK for Learning”, where the former utilizes concept mapping only as an organizational tool, and the latter positions concept maps as artifacts to be assessed in order to guide students’ learning process. In either case, the concept map is not used to determine the extent of a student’s learning, but rather as a part of the learning process. In this sense, concept maps are intended to aid the student in contextualizing new information within their existing body of knowledge [128]. When concept maps are used in this way, students are better able to recall and apply the knowledge gained [49, 87].

The effectiveness of this technique, however, is limited by the ability of students to easily and quickly create concept maps. One of the major hurdles to adoption is that creating concept maps by hand is a difficult process, since it is easy for students to run out of room or allow aesthetic considerations to influence their maps [134]. Techniques have been suggested to simplify this process (e.g., using post-it notes for concepts and moving them around a whiteboard), but the more common solution is to use computer-based concept mapping tools [128]. Weinerth et al. list several advantages of computer-based concept mapping over pencil-and-paper, and the item at the top of the list is the ease of creating and modifying maps [132].

In addition to the benefits to be gained simply by constructing concept maps as part of the learning process (“SAK as Learning”), the creation and modification of concept maps throughout the learning process can provide instructors with insight into the current thinking of their students, allowing for more rapid correction of misconceptions and reiteration of poorly-understood concepts (“SAK for Learning”) [128]. Weinerth et al., Hu et al., and Ho et al. all make mention of the importance of feedback in the learning process as well [49, 132, 134]. It is possible to simply highlight differences between the student map and the expert map, but this requires a concentrated effort on the part of the student to reflect on and examine the reasoning behind the displayed differences. A more effective method is to provide specific targeted feedback for each student map [128]. There is some research into automatic generation of feedback based on ex-

pert maps [134], but for the most part, this requires detailed work to associate specific feedback with each component of the map [87, 132]. Wu et al. concluded that in their study the use of the “evaluation-feedback-modification cycle... significantly improved the learning achievement of the students” [134]. Trumpower and Vanapalli similarly concluded that students who used concept maps as a learning tool performed better in traditional assessment tasks, and when surveyed, found them to be beneficial [128]. The studies we examined typically found concept mapping as an educational tool to be beneficial, but there is a significant caveat to the use of computer-based tools for the construction of concept maps.

The construction of concept maps is a complex task, since it aims to represent a student’s understanding of many related concepts, so it is critical that the software used not increase the complexity of this task [132]. This is especially important in the area of assessment (i.e., SAK of Learning) since an assessment instrument cannot effectively measure comprehension of a subject area if it is instead measuring competency with the instrument itself. Despite the importance of this area, Weinerth et al. found that very few papers addressing assessment using concept maps make any mention of software usability, either directly or indirectly. In the area of assessment, the International Test Commission has requested that usability of assessment instruments be included in all assessment research [132]. However, in the area of pedagogy, these usability findings could be considered more optimistically. Given that poor usability can only decrease the utility of concept mapping software as a pedagogical tool, the positive findings of the papers we examined [49, 87, 106, 134] could only be improved with a renewed focus on software usability in this area.

## CHAPTER 3

### AN ONLINE ENVIRONMENT TO COMPARE STUDENTS' AND EXPERT SOLUTIONS TO ILL-STRUCTURED PROBLEMS

While comparing a student's map to an expert's map can assist with the evaluation, this is a challenging process, in part due to variations in language, resulting in the use of different terms for the same construct. The first step of the comparison is to address these variations by aligning as many of the students' terms with their equivalent in the expert's map. We present the design and implementation of software to assist with the alignment task. The software improves on previous work by optimizing usability (e.g., minimizing the number of clicks to create an alignment) and by leveraging previous alignments to recommend new ones. In addition, alignments can be done collaboratively, as our system is available online: one instructor can invite others to edit or see the alignments. Further improvements to this system may be achieved using content-based recommender systems or natural language processing. The alignment is used as preprocessing to perform the task of comparing maps.

This chapter was published in the following article [38]:

- **Vishrant K. Gupta**, Philippe J. Giabbanelli, Andrew A. Tawfik. An online environment to compare students' and expert solutions to ill-structured problems. In *Proceedings of the Human-Computer Interaction Conference (HCI 2018)*, to appear, 2018.

My contributions consisted of developing the software and defining its architecture, database, implementing and testing it.

### 3.1 Introduction

Problems in classroom contexts are often *well-structured*; that is, they have predefined answers. Assessing a student's work thus consists of contrasting the student's answer to the set of valid answers. However, the problems faced by practitioners are frequently *ill-structured problems* (ISPs), as they admit a much broader set of solutions, which are often contextual [67]. For instance, we may need to identify a solution to improve employee morale. A broad set of solutions exist, including improving shared governance, pay raises, or additional benefits. These solutions are context-dependent, as the specific legal and economic constraints of a workplace may limit the applicability or shape the effects of a solution. Evaluating the ramifications of each possible solution thus requires a deep understanding of the overall system, that is, an appreciation of the relevant variables and their interrelationships. To solve an ISP, student must first form a mental representation of the system by integrating the causal mechanisms of relevant variables (problem representation). Then they can evaluate the consequences of possible actions based on this mental representation (solution generation).

Although theorists suggest ISPs in the classroom as one way to better support problem-solving skills, instructors lack scalable and reliable technological solutions to assess a student's solution in the form of a mental model. The situation is further complicated as students differ not only in their understanding, but can vary tremendously in the terminology that they use. This assessment challenge limits the use of ISPs in the classroom [111], thus precluding learning outcomes and problem-solving skills for students [84].

One way to assess student learning is through the evaluation of their mental models. A mental representation can be externalized in the form of a *causal network*, which articulates antecedents and consequents (Figure 1.1a). Many software tools allow students to generate causal networks (e.g., *Coggle.it*). As causal network generation tools have emerged, assessment tools are also

becoming of interest to educators [32, 56]. Given the open-ended and context-sensitive nature of ISPs, a student's causal map can be assessed by contrasting it with an expert's map. Indeed, studies on problem solving have revealed that expert and novice understanding differ in several ways (e.g., conditional rules [48] and multiple causal paths [58]). While this provides a clear rationale to contrast causal networks, a perfect comparison is computationally infeasible. For instance, finding the largest part of a student's network that matches the expert's cannot currently be accomplished in a reasonable amount of time, as this is known in computer science as *subgraph isomorphism* and is an NP-complete problem [8]. The comparison becomes feasible when the networks have unique concept names [16], but terms in the students' networks are unlikely to be exactly the same as the expert's (Figure 1.1).

To assist instructors in assessing students' solutions to ISPs, we recently developed software which mainly differs from other solutions through its focus on alignments between the terms used by a student and the expert (Figure 1.2) [32]. The software, Incremental Thesaurus for Assessing Causal Maps (ITACM), allows instructors to align each term of a student's map with a term in the expert's map. This is critical from a learning perspective to address the variability in terminology that students use. Alignments are saved on the instructor's computer, and are re-used to expedite the alignment of the next student's map. For example, if the first student had a term "increased salaries" which was aligned to the expert's "pay raises", and the next student also has the term "increased salaries", then it will be automatically aligned with "pay raises". In other words, the process of aligning maps gradually generates a 'thesaurus' which simplifies future alignments. The process nonetheless had three shortcomings. First, the process creates high cognitive demands and relies on extensive human-computer interaction: for each term of the student's map, instructors must scroll through the long list of all of the experts' terms to find if any constitutes a good alignment. Second, an instructor would not benefit from the thesauri created by colleagues in the same field, unless they knew of each other and exchanged thesauri (e.g., by email). Third, even in the same field, there may be multiple valid alignments.

In this chapter, we present a new version of the software which makes the alignment process efficient and informative by addressing all three shortcomings as follows:

- We use interactive visualizations to support each instructor in quickly aligning terms by minimizing the of clicks.
- We create an online environment in which assignments can be shared, so that alignments become a collaborative effort.
- We make it possible to leverage an instructor's community of practice to recommend candidate alignments (based on the alignments of others and their specific educational fields), which reduces the need to scan through a long list of candidates.

In section 3.2.1, we briefly highlight the importance of assessment in education, and the types of computational approaches that have been explored. Section 3.3 introduces the key functionalities of our software and details its architecture. As interactive visualizations are essential in our approach, section 3.4 shows three classical examples that can be used to align terms. We focus on our new alignment process, by explaining the interactive visualization and the recommendation process. Finally, section 3.5 discusses the implications of this work for assessing ill-structured problems in education, as well as benefits to other fields. To support transparency and reuse, readers can connect to <https://osf.io/jr42t/> and obtain the source code, installation manual, and ready-to-use software.

## 3.2 Background

### 3.2.1 Computational approaches to assessment

Knowing the trajectory of a student or automatically analyzing a submission are essential tasks to provide students with appropriate assessment and feedback. Consider that the instructor has a thoroughly researched mental model ('expert map'), and that a handful of students have submitted their mental models ('student maps') by drawing on paper or in a computerized format using `Coggle.it`. In this case, the instructor may spend one or two hours manually assessing the students' maps, by comparing them with the expert map based on a rubric (e.g., 1 points for every equivalent causal connection found). In contrast, a large volume would rule out the possibility of performing a manual assessment.

Computational approaches have thus emerged to offer solutions in various challenges related to assessment, from the broadly defined notion of "is the student on a path towards success" to our more specific challenge of "how can I evaluate the mental model of a student". The application of these approaches is most visible in Massive Open Online Courses (MOOCs), where assessing the learning experience of a large number of participants is an ongoing challenge [117]. MOOCs have extremely high attrition rates [78], with several studies finding that less than 10% of students completed a MOOC [43, 71].

Regression is employed in many statistically-oriented studies. For instance, Sher has employed stepwise regressions to understand what variables contribute to student learning and satisfaction. After controlling for variables such as age, gender, or Internet experience, he found that interactions with students and instructors were significant contributors [112]. Our focus is on computational rather than statistical methods. We do not intend to provide a survey of the field, but rather to highlight its breadth and situate our contribution. For more comprehensive surveys, we instead



refer the reader to Romero and Ventura's 2010 review on educational data mining [102], which articulates over three hundred references, or to the 2016 overview of practices in the field [109].

Computational approaches are particularly popular for building predictive models, which entails methods such as *classification*. Classification is a supervised machine learning method. Specifically, it consists of providing an algorithm with a training set of data containing characteristics and outcome(s) of interest (e.g., students' prior experience with online learning and whether they succeed or not), from which a model is automatically inferred to predict the outcome given only the characteristics (e.g., given a student's previous experience, will he/she succeed?). A plethora of studies have been conducted for classification in MOOCs [2, 83, 103]. For instance, He et al. used classifiers for predicting whether a student was going to complete a course [39]. A wide variety of algorithms have been used for classification. As the best algorithm may not be known *a priori*, studies can use and contrast a variety of algorithms such as decision trees, random forests [60], neural networks, support vector machines [73], or even new algorithms developed specifically to learn from behavioral data [94].

While classification often (but not necessarily) examines the features of a student in isolation, other techniques draw our attention to students as part of a community of learning. In contrast with the supervised approach of classification, which uses outcomes to guide the search for patterns in the data, *clustering* is an unsupervised method which looks for similarities between groups based on features. Clustering has been applied to many studies in MOOC research. For example, Kizilcec and colleagues identified groups of students based on course engagement in the Coursera platform [77], and Ferguson & Clow later took this analysis to the FutureLearn platform [23]. The students are not necessarily the object of clustering, as shown by the study of Ezen-Can et al. who instead clustered forum posts [22].

Similarly to cluster analysis, network analysis provides tools to study students as part of a community. While cluster analysis may see two students who have never interacted as part of the same group based on features and learning trajectories, network analysis explores patterns of

interactions. Contemporary learning theory contends that knowledge is constructed through interactions [65, 118]. Several studies have analyzed the structure of interactions, that is, who a student interacts with and when. Studies have found that the dropout rate is reflected in the structure of the network formed from students' interactions as it becomes sparser and increasingly fragmented over time [33, 34]. Our previous analysis also found that interactions were heavily dependent on a handful of highly-engaged students [124]. The centrality (i.e., structural importance of a student within the network) has also been investigated, with mixed results [17, 62]. To analyze who a student interacts with, studies can examine online forums or chats. These offer textual information, which can be mined to assess how phases of learning (e.g., integrating others' perspectives, clarifying questions, or articulating one's own thoughts) relate to interactions [123].

The concept of a network has broad applicability, as it is not limited to *social* networks formed by interactions between students or between students and instructors. For instance, Dascalu and colleagues took a network approach to analyzing the structure of online discourses in the *Reader-Bench* framework [14]. In another context, Hecking et al. examined how students accessed online resources, which was represented as a bipartite student-resource network [40]. Our focus is on the use of a network to assess a student's work, which is a necessity for instructors as well as the students themselves. As recently noted by Schumacher and Ifenthaler, students in online learning environments expect to be given tools for self-assessments, among other analytical features, to self-regulate their learning [108]. The thematic working group at the 2015 EDUsummIT also emphasized the need for instructors to provide formative assessments and feedback to enhance student learning [117]. The approach of comparing a student's map with an expert's map, which is central to this work, was particularly underscored by the working group which highlighted the HIMATT and AKOVIA tools [56]:

The most promising recent advances in... formative assessment for complex learning tasks involve a series of research efforts... consolidated in Highly Integrated Model

Assessment Tools and Technology (HIMATT). HIMATT provides a learner with a problem situation and then prompts the learner to indicate (in the form of text or an annotated graph) the key factors and their relationships involved in addressing the problem. This problem conceptualization can be compared to an expert conceptualization or reference model and analysed to indicate things for the learner to consider. [117]

### 3.3 Software functionalities and architecture

#### 3.3.1 Overview

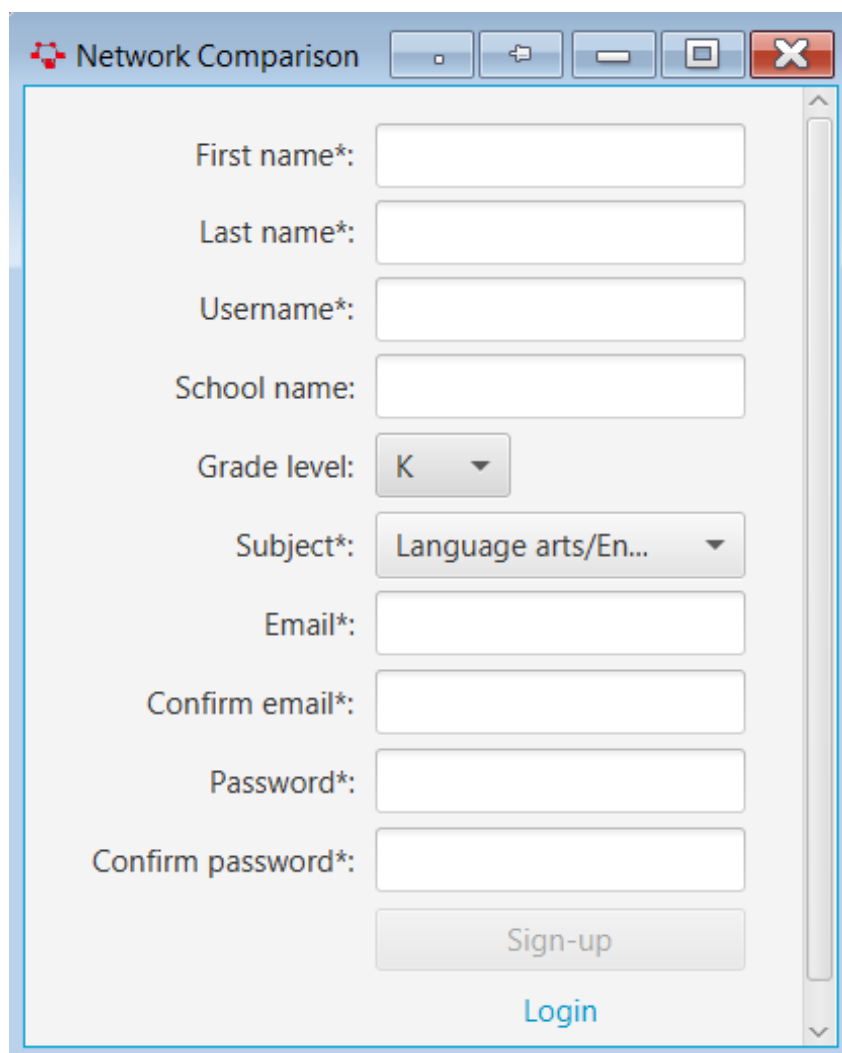
When detailing interactions with our software, we refer to instructors as *users*. As in any online application, the full application does not reside entirely on the user's side. Rather, the user has a lightweight graphical application (a *client*) to request services, such as 'create an assignment' or 'recommend terms to align', and these services are accomplished through a university-hosted computer (the *server*). A user thus starts by downloading and running the client.<sup>1</sup> For the first use, the user needs to sign up by providing information including a username, email<sup>2</sup> and password, and subject taught (Figure 3.1). The subject is important, as it allows contextualizing language when generating recommendations for alignments from the system. Immediately after signing up, the user can log in.

Once logged in, the user can create a new *assignment* to host a set of maps. For instance, if the user has an in-class activity where students produce maps, then an assignment would be created

---

<sup>1</sup>The client communicates with the server via the http protocol on port 8080, which is commonly used to offer web services. Our application thus requires a single, common port to be open on the server-side. Note that port 80 can also be used for http. However, it is common for this port to be already used by an application, thus we use 8080 as a default.

<sup>2</sup>While we check that the email is entered in a valid format, we do not send a confirmation email to register. A university-hosted computer with the ability to act as an email server (even for very specific emails) poses greater security risks. To simplify the deployment of our software, there is thus no need to install an email server or to obtain security clearance for this purpose.



The image shows a window titled "Network Comparison" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a sign-up form with the following fields and controls:

- First name\*:
- Last name\*:
- Username\*:
- School name:
- Grade level:
- Subject\*:
- Email\*:
- Confirm email\*:
- Password\*:
- Confirm password\*:

At the bottom of the form, there is a "Sign-up" button and a "Login" link.

Figure 3.1: Signing up to create an account.

to indicate that these maps all originate from the same activity. All fields are sortable. To help instructors in further organizing their content, we thus have optional (sortable) fields such as the number of points and the due date (Figure 3.2).

In order to support interaction with their community of practice, users can collaborate both directly and indirectly. An indirect collaboration occurs when a user receives a recommendation for an alignment based on the work of many other anonymous users. A direct collaboration occurs when a user explicitly wishes to work with a specific set of other users for a particular assign-

ment. Upon the creation or editing of an assignment, a user can specify who to share it with by entering their emails one after the other. They can contribute through the maps, for instance by visualizing or creating alignments. This allows them to create *communities of practice*, thus supporting instructors through the difficult task of preparing the implementation of ill-structured problems. Permissions can be revoked by clicking on the button next to a user with whom the assignment was previously shared. After sharing, the original user who created the assignment remains its owner. In contrast, users with whom the assignment was shared are not able to edit the assignments' properties (including sharing) or to delete it.

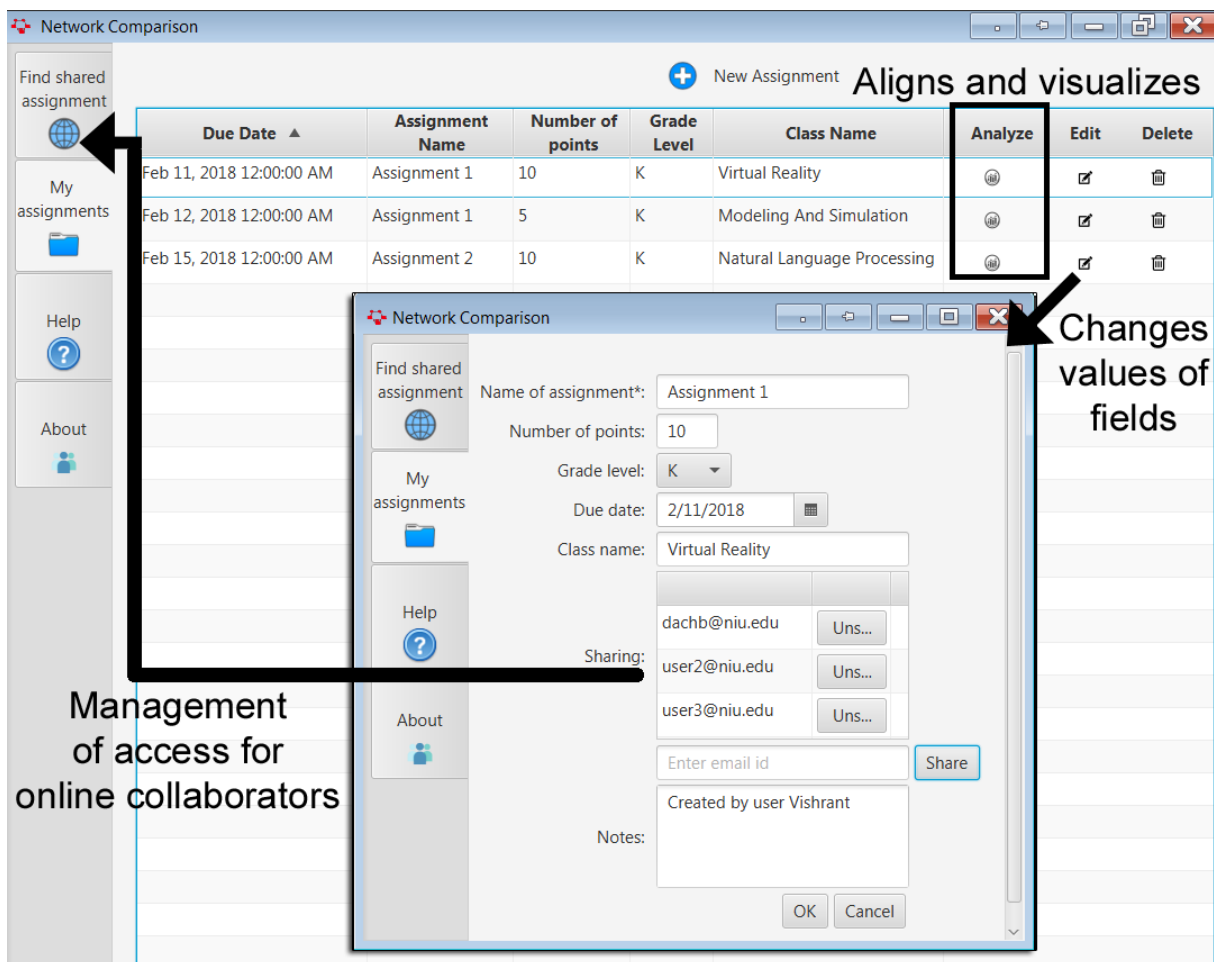


Figure 3.2: Creation of a new assignment (foreground), which allows adding and revoking sharing permissions with other users. The user has three assignments (background), currently sorted by due date.

Clicking on the analysis button (available both for the users' owned and shared assignments) opens a new window where each analytical task is hosted in a separate tab. Regardless of what tab is activated, a user can add student maps by uploading them onto the server, or remove them. Similarly, the one expert map can be added or removed. Note that only the maps are uploaded: we do not store confidential meta-data such as the author of each map. This would complicate the sharing mechanism, as a user may elect to share the students' work but hide the students' identity. If a user chooses to disclose the authors for each map, then the map files can simply be named after their authors.

The first tab, which opens by default, is devoted to the alignment process (Figure 3.3). The second tab allows interactively exploring and contrasting two maps, displayed using node-diagram representation (Figure 3.4). That is, each variable in a map is represented as a labeled circle (*node*), and the fact that a factor  $A$  is a logical antecedent for  $B$  is represented as a directional arrow from  $A$  to  $B$  (*edge*).

### **3.3.2 Architecture and technologies**

From an educational or application perspective, we refer to the work of each student as a 'map'. From an analytical perspective, the same notion is generally named a 'network', whereas in computer science it may be called a 'graph' (particularly when the focus is on storing and accessing its information). Given the focus on implementation in this sub-section, we will use the term 'graph'.

As detailed in the previous sub-section, our software is built on a client-server model. The technologies involved in the client are summarized in Table 3.1. The client is programmed in Java. When the client selects a map to upload, we attempt to convert it into a graph object on the client side using the `Jung` library. If this succeeds, then we conclude that the file is properly formatted,

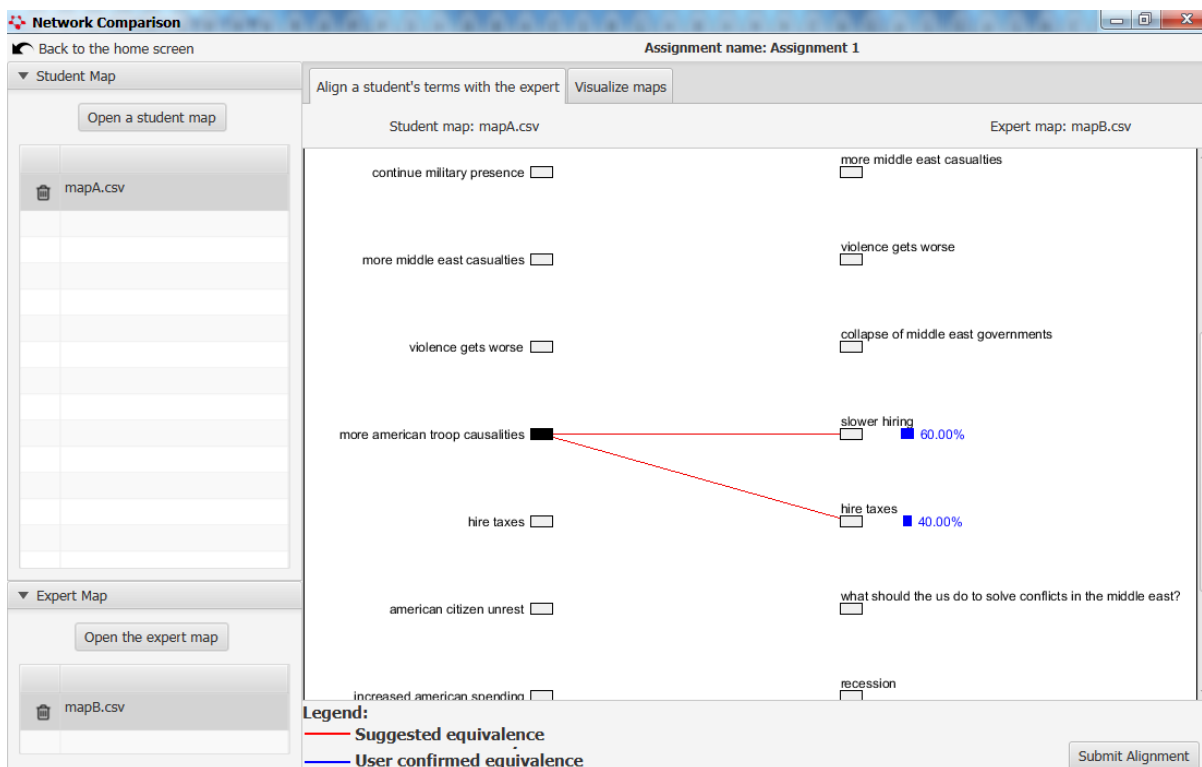


Figure 3.3: Alignment of terms from a student (left) and the expert (right).

and we allow the client to send it to the server. The client uses the `GraphStream` library to visualize the alignment, and `D3` together with `D3plus` to display the graph in the visualization tab (Figure 3.4).

The server is rooted in Spring's web Model–View–Controller (MVC) framework. The architecture is shown in Figure 3.6. In our application, developing using the MVC framework leads us to need a combination of multiple technologies, which is reflected in the extensive list of server-side libraries (Table 3.2). Overall, the controller layer takes in requests from the client, handles them, and returns a response. Specifically, when a request comes in from a user, the dispatcher hands it over to the handler, which inspects and processes the request by sending it to the next layer (a service). The service will perform the request and return the answer to the controller, which sends it back to the user. Apart from signing up, all other services require authentication to access resources. Thus requests are further sent to the authentication manager within the authentication

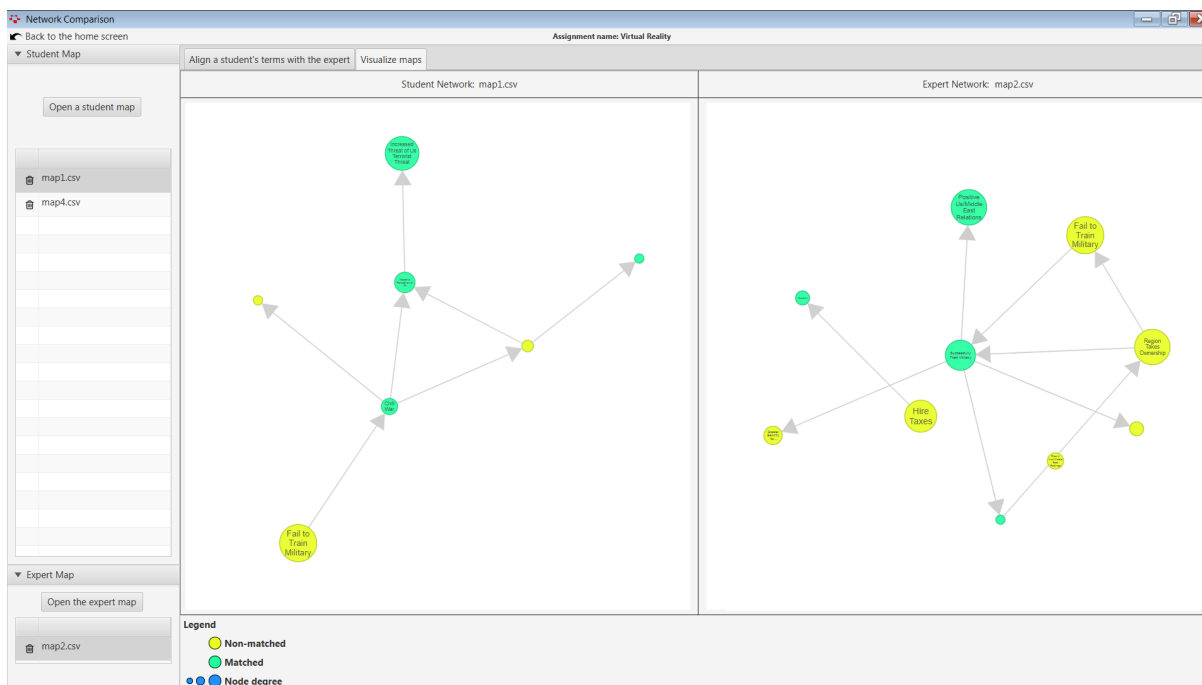


Figure 3.4: The visualization shows what factors have been aligned, and which ones are important in number of connections using betweenness centrality.

layer. The authentication manager produces an authentication token for valid credentials. In line with the standard RFC 7519, we have used JSON Web Tokens. The Spring Security Framework ensures that each authenticated request runs in a `ThreadLocal`, which means that it can only be read and written by the same thread.

The web service provides interoperability between systems. The Representational State Transfer (REST) architecture is a commonly used technology for web applications, as it is lightweight and scalable. Services built on a REST architecture are known as RESTful. We used Spring to implement a RESTful web service.

Finally, the last layer contains the databases. We use two different databases, a Relational DataBase Management System (RDBMS) based on the open source MySQL (the second most widely used database engine in the world), and a graph database management system (Graph DBMS) based on Neo4J (the most popular graph DBMS as of February 2018 per



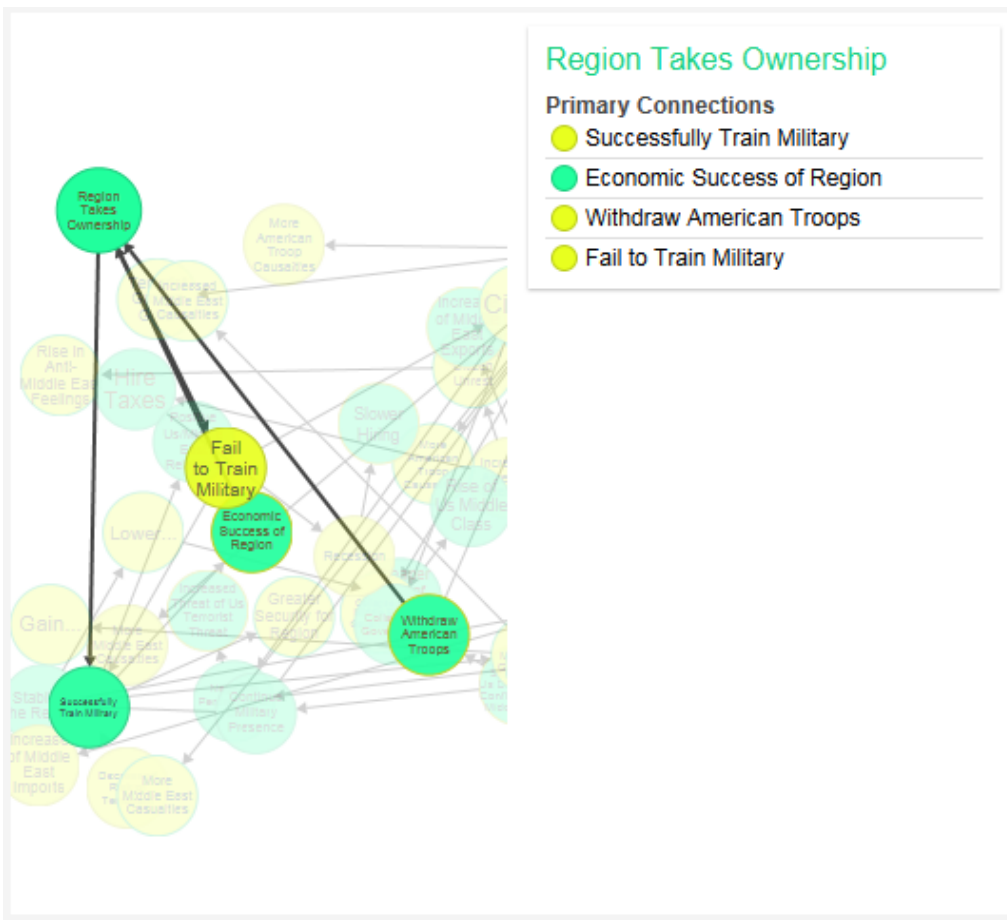


Figure 3.5: Interactive visualization of connected concepts implemented using the D3Plus JavaScript framework.

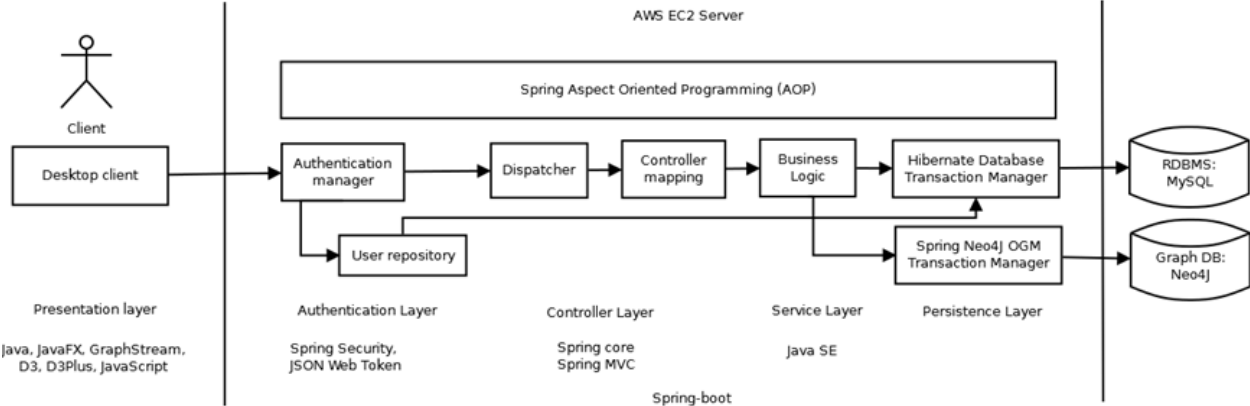


Figure 3.6: Model, View, Controller (MVC) based N-tier system architecture.

Table 3.1: Client side architecture

Architecture layer	Presentation
Components	Desktop client
Technologies	D3 3.5.15 D3plus 1.9.8 Java 1.8 Jung 2.0.1 GraphStream 1.3

<https://db-engines.com/en/ranking>). The MySQL database records all individual characteristics. That is, we have the characteristics of each assignment, each user, each file, and the alignment made by each user for each file within each assignment. In contrast, the Neo4J database records only aggregate characteristics consisting of the number of times an alignment has been endorsed by users of a specific type.

There are two broad approaches to interacting with a MySQL database: we can either write SQL code or automatically transform JavaBeans (i.e., classes encapsulating multiple Java objects) into database operations. While solutions such as the Java Database Connectivity API (JDBC) allow one to communicate using SQL code, its behavior depends on the choice of an RDBMS, which makes it difficult to migrate the application from one database system to another. In contrast, an automatic transformation removes the dependency on a database-specific query language. We thus opted to use the Hibernate Object/Relational Mapping (ORM) framework, which generates queries (e.g., column mapping and joins) based on Java classes and table configurations provided through the Hibernate framework annotations. Similarly to Hibernate ORM, we transform Java classes to interact with the Neo4J database using the Neo4J Object-Graph mapping library.<sup>3</sup>

---

<sup>3</sup>Hibernate could be used for *both* SQL and Neo4J, with Hibernate ORM and Hibernate OGM respectively. Opting for this solution would require changes in the overall architecture, as it would start to resemble a microservice architecture.

Table 3.2: Server side architecture.

Architecture layer	Controller	Authentication	Service	Persistent
Components	Dispatcher Handler mapping	Authentication manager. Uses: - user repository - user service	Web service	SQL Database NoSQL Database
Technologies	Spring boot 1.5.9 Spring core, MVC, and AOP 4.3.13	Spring boot 1.5.9 Spring security & Web 4.2.3 JSON Web Token 0.6.0	Java 1.8 Spring web 4.3.13 Spring boot 1.5.9	Spring boot 1.5.9 Hibernate 5.0.12 MySQL 5.7.14 Neo4J Java 3.1.4 Neo4J DB 3.2.3

### 3.3.3 MySQL database

The MySQL database is used throughout the application to create assignments, add files, and directly collaborate with other users. We thus detail its schema as part of the general software architecture. In contrast, the graph database is used only for alignments, and is discussed in section 3.4.2. Information in SQL is structured in relational tables, where each row of a table has a unique identifier (known as the ‘primary key’). These identifiers are automatically set (i.e., incremented) by the MySQL database and are not discussed below. Similarly, an entry may include an identifier from another table (e.g., to know which user owns an assignment). These ‘foreign keys’ are represented as pink diamonds in Figure 3.7 and will also be omitted from the discussion.

The MySQL database involves six tables (Figure 3.7). The content of the `Appuser` table was provided by the user when signing up (Figure 3.1), except for the last field (‘authorities’). This field is required by Spring Security, and is systematically set to `ROLE_USER` in our application. Each user can create many assignments. All of the information in the `Assignment` table is provided through the edit window (Figure 3.2). Sharing is a special case as it is formed of a *list* of users rather than of a single value (e.g., single due date, single class name). Rather than storing a list of values within a single cell of the table, best practices in database normalization encourage the

creation of a distinct table. This is further complicated by the fact that some of the users with whom we wish to share (e.g., `user2@niu.edu` in Figure 3.2) may already exist and can be given immediate access to an assignment, while others may not have registered yet. We resolve this situation using two tables: `Shared` for users receiving immediate access, and `Invited_users` for those who will get access if they choose to register.

The `Files` table tracks all the files used in a given assignment. It records the name of the file as it was uploaded on the server, whether the file is a student or expert map, and it binds it to a JSON file. The JSON format is used as our file format to store maps. Our experiments showed that it leads to better loading times for users compared to keeping it in the `Coggle.it` format.

The `User_Alignment` table tracks how a specified user has aligned the content of specific files in a specific assignment. When a user opens an assignment, this table is used to display the alignment that he/she has made previously. If the user wishes to get recommendations for new alignments, then we will query the Neo4J database. If the user attempts to update alignments, we first check in the `User_Alignment` data that the alignments are indeed different, and we only then update the Neo4J database. Since the Neo4J database only tracks aggregate alignments and not individual users, allowing one user to update alignments  $n$  times *without* checking that they are different would be erroneously recorded in Neo4J as alignments produced by  $n$  users, thus biasing recommendations.

## 3.4 Alignment process

### 3.4.1 Visualization

To create an alignment in the first version of ITACM (Figure 1.2), users had to (i) click on a drop-down menu to open, (ii) scroll through the list, (iii) click on the term to align with, and in general (iv) click to close the menu since it hides the access to the next menu. One of our

contributions is to use interactive visualizations that minimize the number of clicks necessary to align terms. Our visualization is similar to `Jigsaw` (section 3.2), although `Jigsaw` was made to display connections *found* in a collection of documents and not to *create* these connections.

An alignment is defined by a student term and an expert term: a minimum of two items thus have to be selected. Our new visualization achieves this minimum of two clicks. The user clicks on a student term on the left (e.g., ‘more american troop casualties’ for the student in Figure 3.3), and then clicks on the expert term to align. Once the user has made the alignment, it appears in blue. To move to the next alignment, the user again clicks on a student term.

To minimize the cognitive load when performing an alignment, it is important to also minimize the time spent scanning the list of possible candidate terms on the expert side. That is, given a student term, a user needs support to identify likely alignments. Our recommendation algorithm is detailed in the next section. From a visualization standpoint, recommended or ‘candidate’ alignments appear in red (Figure 3.3). This helps the user to immediately identify relevant terms, and it differs from the user’s alignments which appear in blue. Similar to `Jigsaw`’s use of a bar next to the term, we use a labeled bar that shows the likelihood that each term is a good alignment. For example, Figure 3.3 shows that ‘slower hiring’ is the most likely term at 60% whereas the other candidate is at 40%.

### 3.4.2 Aggregate recommendation

There are three key principles to recommend an alignment. An alignment will be recommended to a user proportionally to the number of (i) *other* users that endorsed it (ii) *exactly*, while (iii) *favoring* users from the same application context. There are two consequences of this alignment approach. First, a user will not get a recommendation from his/her own alignment. Consider for instance that user  $U$  believes that “money” maps into “happiness”, but no other users have maps including such terms. If  $U$ ’s alignment was used, then  $U$  would be told that the alignment is right

since it occurs in 100% of applicable cases. Our exclusion thus avoids self-confirmatory biases. Second, even the slightest difference between terms means that they are treated as different, and the alignments that they are involved in are thus also different. For example, a user may have aligned “stressed” to “depression” and another aligned “stressed” to “depressed”. These will be considered as two wholly different alignments. This is a limitation of our approach, as we do not currently use Natural Language Processing. The third principle reiterates the importance of contextualizing language. If a user is trying to align terms for a music assignment, we will give more weight to alignments also made in the context of music than in other contexts. That is, we use a factor  $\alpha > 0$  which increases the weight of a recommendation made in the same context as the user’s assignment.

We start by detailing the simplest case where a user has made no alignment. This will demonstrate principles (ii-iii) and omit (i), which we will discuss as part of the general case. Our guiding example is a user, Jane, working on a music assignment. The state of the database is shown in Figure 3.8(a) while the user’s list of student and expert terms are shown in Figure 3.8(b). Assume that Jane clicks on the student term ‘note’. This prompts our algorithm to retrieve the list of alignments involving the exact term ‘note’ from our Neo4J database, thus implementing principle (iii). We denote this list of alignments by  $\mathbb{A}$ . Each alignment  $a \in \mathbb{A}$  is annotated with the number of users  $users(a, d)$  that have constructed it, for each application domain  $d \in \mathbb{D}$ . In Figure 3.8(c), we see two alignments: the top alignment is endorsed by 1 music user, and the bottom one by 1 science user. Formally:  $\mathbb{A} = \{\text{Note} \rightarrow \text{Pitch and duration}, \text{Note} \rightarrow \text{Grade}\}$  with  $users(\text{Note} \rightarrow \text{Pitch and duration}, \text{music}) = 1$  and  $users(\text{Note} \rightarrow \text{Grade}, \text{science}) = 1$ .

We use a preferential weight of  $\alpha = 1.5$ , which means that alignments made by other music users are worth 50% more. The total weight  $\mathbb{W}(A)$  of the alignments, taking into account  $\alpha$  and the user  $u$  to whom we made the recommendation is as follows

$$\mathbb{W}(A) = \sum_{a \in \mathbb{A}} \sum_{d \in D} \begin{cases} users(a, d) & \text{if } d \neq domain(u) \\ users(a, d) \times \alpha & \text{if } d = domain(u) \end{cases} \quad (3.1)$$

In our guiding example,  $\mathbb{W}(A) = 1 \times \alpha + 1 = 1 \times 1.5 + 1 = 2.5$ . The weight  $w(a)$  of an alignment  $a \in \mathbb{A}$  is the weighted summation of the users having this alignment, normalized by the total weight of the alignments:

$$\frac{\sum_{d \in D} \begin{cases} users(a, d) & \text{if } d \neq domain(u) \\ users(a, d) \times \alpha & \text{if } d = domain(u) \end{cases}}{\mathbb{W}(A)} \quad (3.2)$$

Here, ‘Note  $\rightarrow$  Pitch and duration’ has a weight of  $\frac{1 \times \alpha}{2.5} = 0.6$ , while Note  $\rightarrow$  Grade has a weight of  $\frac{1}{2.5} = 0.4$ . Both are recommended since they were in the list, but aligning ‘Note’ with ‘Pitch and duration’ has a higher recommendation in the context of music than aligning it with ‘Grade’ (Figure 3.8-d).

If Jane clicks to make the alignment ‘Note  $\rightarrow$  Pitch and duration’ and then commits it, the database changes: this alignment is now done by 2 music users, and 1 science user. *Another* music user, George, would thus see a weighted total of alignments of  $2 \times 1.5 + 1 = 4$ , and be told to align ‘Note’ with ‘Pitch and duration’ with a weight of  $\frac{2 \times 1.5}{4} = 0.75$  or to align it with ‘Grade’ with a weight of  $\frac{1}{4} = 0.25$ . However, the weights should not have changed for Jane since her own alignments should not bias the recommendations. This leads to the general case.

In the general case, we also need to exclude a user’s alignments when computing recommendations for this user. To achieve this,<sup>4</sup> the list of alignments  $\mathbb{A}$  is modified by decrementing  $users(a, d)$  when (i)  $a$  has been chosen by the target user, *and* (ii)  $d$  is of the same domain as the user’s. For instance, our music user Jane has done the alignment ‘Note  $\rightarrow$  Pitch and duration’, which has now a score of 2 for music users. We would thus update this score to  $2 - 1 = 1$ . If the same alignment had been scored by science users, then this score would be unaffected.

### 3.5 Discussion

Comparing the map of a student and an expert is a difficult assessment problem, in part due to variations in language. Identifying when the term used by a student is equivalent to an expert’s term is thus an essential step to eventually compare maps as students solve ISPs. The previous version of our software focused on aligning student terms with expert terms, and it introduced the notion that previous alignments can be used to more quickly find the next ones. The newer version of the software presented here proposes several significant improvements. The alignment process is significantly simpler, by improving both on the usability (e.g., minimizing the number of clicks necessary to make an alignment) and on the quality of recommendations to highlight potential alignments. The process also becomes fully collaborative, as our software is now online and allows us to easily invite other instructors to work or see the results of an alignment. These improvements were enabled by the articulation of several technologies, including structured databases (MySQL), graph databases (Neo4J), and visualization tools (D3).

---

<sup>4</sup>An alternative is to modify the core of equations 1 and 2 into:

$$\left\{ \begin{array}{l} users(a, d) \text{ if } d \neq domain(u) \text{ and } u \text{ did not choose } a \\ users(a, d) \times \alpha \text{ if } d = domain(u) \text{ and } u \text{ did not choose } a \\ users(a, d) - 1 \text{ if } d \neq domain(u) \text{ and } u \text{ chose } a \\ (users(a, d) - 1) \times \alpha \text{ if } d = domain(u) \text{ and } u \text{ chose } a \end{array} \right. \quad (3.3)$$

We did not retain this solution, and instead opted to keep the equations while updating the dataset on which they are applied.



There are several limitations to our approach, which may be addressed in future work. First, our recommendation was based on a weighted aggregation method. This heuristic was used as the data is small, since we are starting the system and we have a very high-dimensional space of potential alignments. However, as data becomes available, content-based recommender systems would be able to make better use of the data to offer recommendations [1]. In addition to recommendation quality, we also need improvements in transparency: our scores indicate the best fit (e.g., 60% is more recommended than 40%), but the user does not currently know how this score was computed and hence why it should be trusted. Gedikli and colleagues have developed “guidelines of how to build explanation interfaces for recommender systems that support the user in the decision making process” [26], and such guidelines are a promising direction to ensure that users do take recommendations into account for the right reasons.

Second, our system will only form a recommendation for a term if it was used without alteration in a previous alignment. A human may look at terms that differ only due to typos, or the use of plural instead of singular, and realize that they are the same term. Our system would consider them as wholly different terms. Natural Language Processing (NLP) could be used in future versions to preprocess the data and map variations of a term into a common root. Most interestingly, NLP is also a promising approach to identify possible alignments, which can complement the use of recommender systems. For instance, lexical databases such as WordNet can identify relations between words. Methods for semantic relatedness can go one step further, by measuring the strength of association between terms. Such algorithms are available as part of the open source `WikiBrain` Java library [110]. As the library is built upon the 267 Wikipedia language editions, it also offers support to identify semantic relatedness in many languages, whereas WordNet is a lexical database for English only.

While the motivation for this work was to support the comparison of a student’s mental model with an expert’s mental model, it also provides a method and software for the problem of comparing mental models in several other applications. In socio-ecological dynamics, we recently found that

the mental model of a stakeholder (in the form of a map) could be a good proxy to understand how the stakeholder expects the system to change [81]. Comparing the mental models of stakeholders thus holds the potential to contrast their expectations about a system's behavior, and to lead to focused conversations on how to steer the system towards a desired goal. In the field of health, both policies for population health and individual perspectives can be represented as maps [18, 30]. Comparing these maps would thus help to understand how policies or individuals differ. Shifts as captured by changes in a map have also been proposed to detect the emergence of a political crisis, and our approach can be used to compare a map at different time steps [27, 93]. Finally, in the field of Modeling & Simulation (M&S), a map is often produced as part of the *conceptual modeling* stage, which identifies key factors and relationships before the model is operationalized (through mathematics) and implemented (in a simulation model). Comparing models at a later stage such as simulation is challenging, as there are differences in implementation and simulation method (e.g., a map can be turned into a Fuzzy Cognitive Map, System Dynamics, or a Network model). Comparing models at the conceptual stage would allow us to focus on the core hypotheses that underlie a model. Previous work has analyzed models independently at the conceptual stage [31], but a paucity of tools was available to contrast models. Consequently, there is an abundance of opportunities to apply our tool in active research areas, and exploring these opportunities would be a fruitful endeavor for future work.

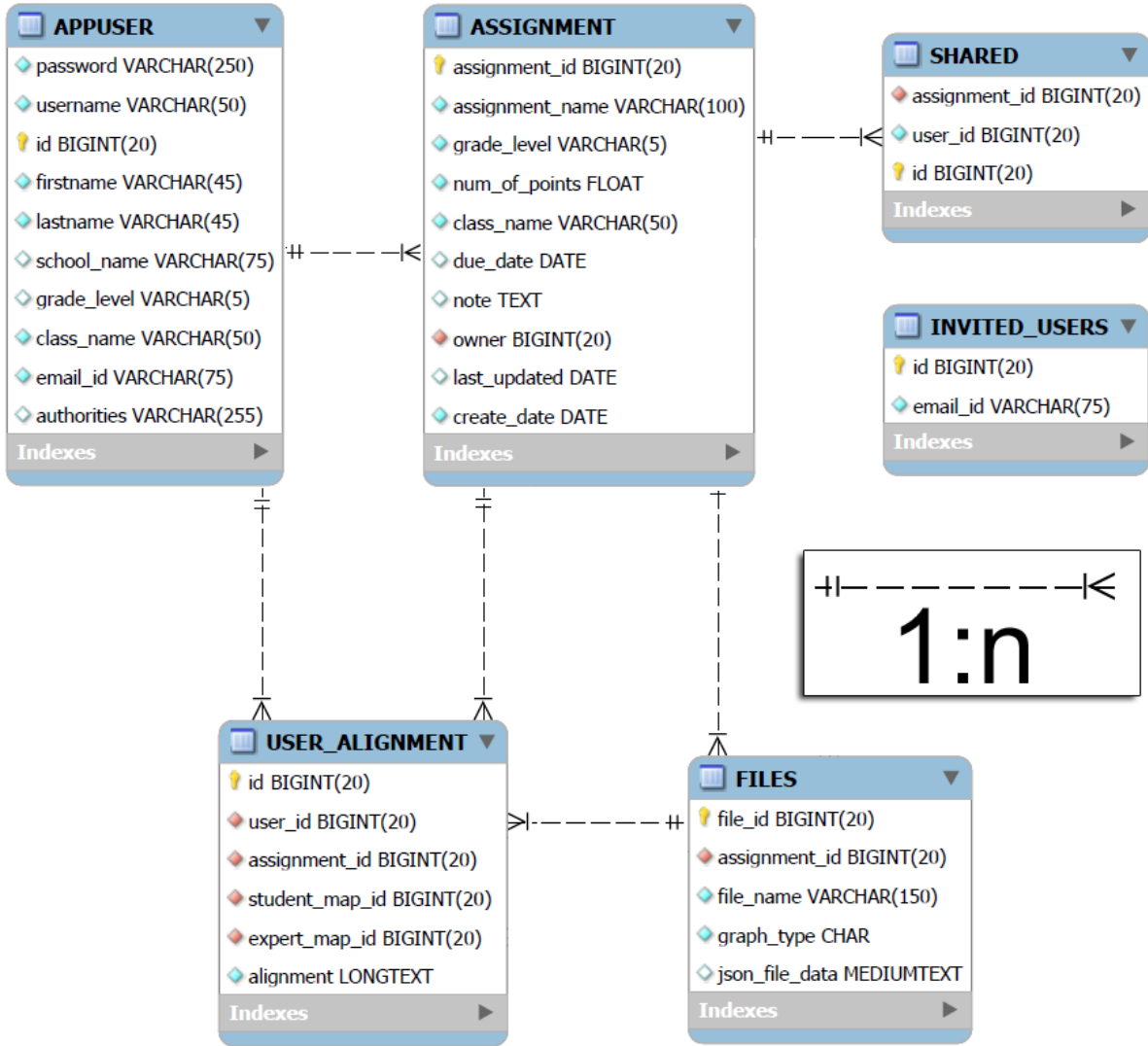


Figure 3.7: Entity Relationship (ER) diagram of the SQL database, from MySQL Workbench.

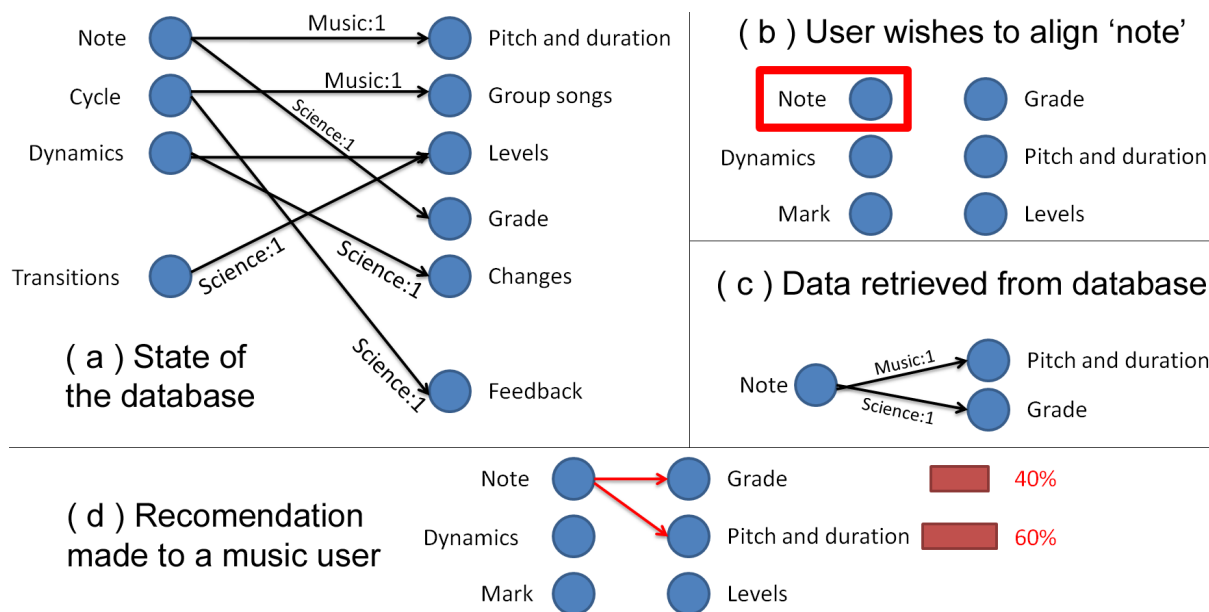


Figure 3.8: Main steps of our algorithm to perform an aggregate recommendation.

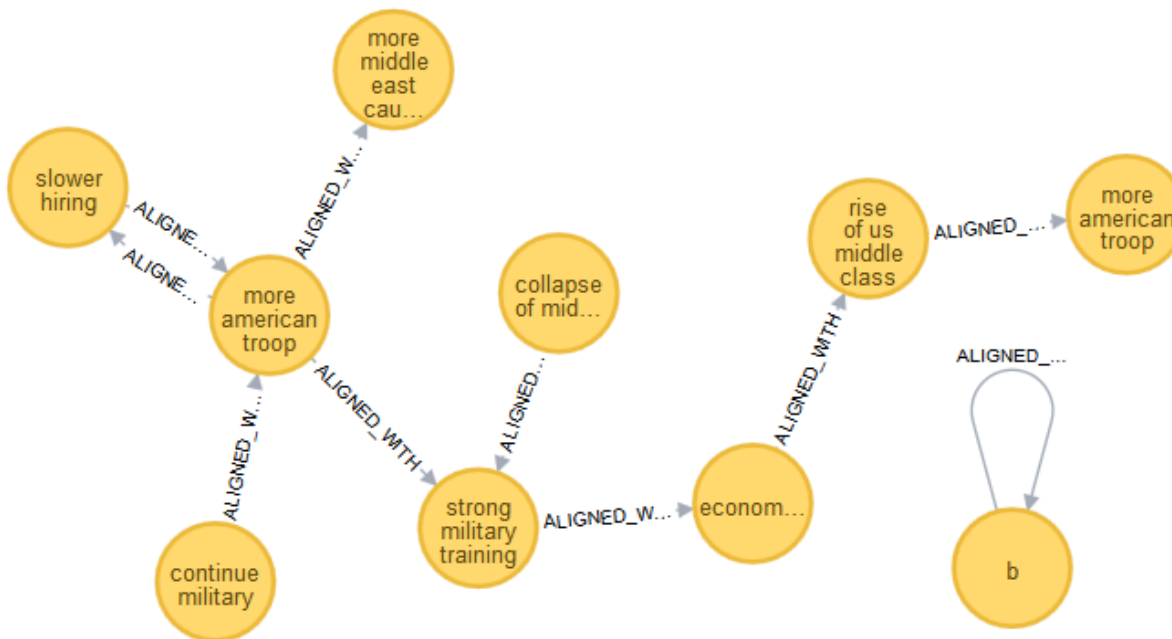


Figure 3.9: Network built in Neo4J for aggregate recommendation.

## CHAPTER 4

### DIGITAL TOOLS TO GUIDE STUDENTS IN ANSWERING ILL-STRUCTURED PROBLEMS

In the first chapter, we set as one of our objectives to assess a student’s map by performing comprehensive network comparisons. The assessment is the focus of this chapter, building on the preprocessing alignment of terms introduced in the previous chapter. Assessing maps can involve a referent-free evaluation (e.g., to encourage the creation of maps with high density of concepts), or a comparison to an expert map used as reference. This chapter starts with a review of theories and tools to compare a student’s map to the expert map. Previous approaches often compared individual connections (e.g., scoring the number of connections that a student has/misses in contrast with the expert) or general map metrics (e.g., one map is denser than the other). In contrast, the problem of comparing two maps has been studied in network theory and graph theory for several decades, yielding categories of algorithms that are currently under-utilized in educational research. This chapter reviews three categories of algorithms (i.e., graph kernel, graph editing distance, graph embedding) in light of their application to assessment and student success. We discuss an implementation of these algorithms through a new set of digital tools, designed to supports a community of practice in problem-based instruction.

This chapter will be included in “Digital tools to guide students in answering ill-structured problems”, a draft being prepared for publication by Philippe J. Giabbanelli, Andrew A. Tawfik and **Vishrant K. Gupta**. My contributions consisted of writing the code for graph comparisons and testing the results by using all three algorithms presented in the chapter.

## 4.1 Introduction

For many years, education has often been administered using didactic and lecture-based methods. In these settings, a teacher often disseminates information and learners are tasked with memorization of information. However, many argue this form of decontextualized education fails to support learning transfer and engender problem solving. To address this challenge, theorists posit that learning should be situated within problem-solving contexts [47, 66]. Instructional strategies that employ ill-structured problems are often referred to as inquiry-based learning. In these instructional strategies, learners are often presented with a problem representative of the types of issues that practitioners face. These problems are often characterized as being ill-structured; that is, the problems lack defined goals or explicit ways to achieve the predefined goal state [42]. In contrast to well-structured problems that have a set of correct answers, an ill-structured problem is often assessed on the viability of the proposed solution given the constraints, perspectives, and standards embodied in a context [135]. The belief is that the ill-structured and problem-solving approach espoused in inquiry-based learning will generate more robust knowledge structures [11, 55, 75].

This shift in education towards an ill-structured approach has implications for theory and practice. In contrast with multiple choice questions that an instructor may use for assessment of a well-structured problem, ill-structured problems require learners to identify the relevant elements of the problem space and generate arguments about why a solution is viable and rational [72]. In addition, Eseryel, Ifenthaler, and Ge [21] see an “effective learning process as one that facilitates transition of problem spaces of learners from the state of preconceptions or misconceptions to the state of comprehensive, causal explanations”. As such, the multiple-choice approach often employed in the information dissemination model fails to meet the assessment needs of inquiry-based learning. Therefore, educators look towards alternative forms of understanding complex problem-solving, such as concept maps, which articulate concepts and their logical antecedents

or consequents, because the creation of these artifacts affords learners opportunities to articulate their understanding of the problem space and the causal relationships between the concepts [53]. Representing the problem space using causal maps is a critical cognitive process not only to the success of that problem solving, but also to the refinement of the student's conceptual knowledge and problem solving skills. In doing so, these mapping processes also help students construct the knowledge acquired into a conceptual framework for that problem [69, 132].

The shift in the student's representation of knowledge creates challenges for assessment. Whereas well-structured problems are assessed by the student's ability to reiterate the predefined answer, representations of solutions to ill-structured problems through causal maps necessitate a varied set of cognitive procedures for assessment. There are two broad categories of procedures. First, a causal map can be assessed through a scoring method that favors certain structures. For instance, Kotovsky and colleagues [79] suggest that one way to understand complexity in problem solving is through the 'number of branches at each node and depth of search to a solution node' (p.248). A metric can thus be developed where a student map with longer paths and more branching would score higher, indicating a more complex map. Such scoring methods are known as referent-free. Structural attributes favored in students' maps have included 'parsimony, temporal flow, total links, connectedness' [61]. Extensions to causal maps have also been proposed, to bring in elements specifically for scoring: weighted concept maps examine the weight of propositions (i.e., links) [10], while a 'concept map+' distinguishes between types of links [91]. Second, a student's map can be assessed by comparison to a reference expert map. Ifenthaler suggested that providing an expert map, and comparing it with student maps (e.g., for model-based feedback), can foster a better understanding of a problem [52]. This potential was confirmed by experimental studies [54, 127]. As pointed out by Trumpower and colleagues, 'hand scoring knowledge maps can be quite time consuming', and many algorithms have been developed to automatically assess digital knowledge maps [126]. These algorithms operate typically at the level of individual nodes and links, to find the ones present in both the expert and the student maps, or present in the expert

map but missing in the student map [126]. In parallel with the growing interest in comparing maps in education, algorithms have been developed over several decades at the intersection of pattern recognition and graph theory to address the related problems of map comparison, network alignment, and graph matching [24, 129]. The uptake of such approaches in educational research has been limited.

The main contributions of this chapter are twofold. First, we detail how approaches in pattern recognition and graph theory (e.g., graph kernels, graph editing distance, graph embedding) can be used to go beyond comparing individual nodes or edges when assessing a student's map using a reference expert map. Second, to benefit the practice in the community of problem-based instruction and the field of education, we implemented these methods through client-server software that supports instructors in comparing maps through several methods.

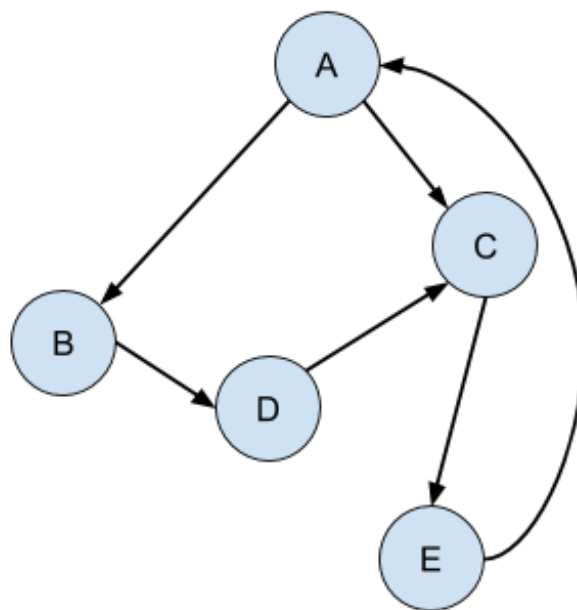


Figure 4.1: A concept map is a network or graph where relevant concepts are captured as nodes (depicted as circles) with logical connections known as links to indicate antecedents and consequents (depicted as arrows).



## 4.2 Background

### 4.2.1 Developing knowledge structures through ill-structured problem solving

Well-structured problems possess all of the necessary information, solution strategies, and criteria for evaluation of the problem. Generally speaking, in these situations problem solving often consists of problem representation and then a search for the correct solution [3]. These problems are often employed in modern education because the predetermined nature of the solution affords efficiency in assessment. Alternatively, inquiry-based learning suggests that learners should be provided opportunities to solve problems that are representative of a domain. Specifically, students' learning is centered around real world, ill-structured problems [82]. These problems provide students with a meaningful real world context for them to structure their domain knowledge schemata in order to effectively retrieve them later in the workplace when needed [7, 107]. Clariana et al. [12] suggest there are five features that distinguish simplistic and complex problem-solving experiences. First, a complex problem often includes a greater number of variables presented in the problem space. Furthermore, the connectivity among the variables and their interdependencies further increase the complexity. They further postulate that the changing dynamics suggest that state change and time are another factor. A further confound in complex problem solving is that a goal state is unclear, also known as intransparency. Finally, several goals coincide, which impact the activity of the individual.

How learners conceptualize the problem space is a critical issue in inquiry-based learning. The problem space not only depicts the major concepts (variables) that have a role in the cause(s) or the solution of the problem, but also provides an underlying explanation for the problem with the causal relationships among the variables. For example, in medical education, the problem

space should “include[s] all the causal mechanisms that account for the patient’s signs and symptoms” [46]; that is, the understanding of the problem is also described by the causal relationships among the variables that detail the mechanisms for why the problem occurs and how it can be solved [21]. Furthermore, when students construct a problem space through causal reasoning, they are practicing the scientific problem solving process and consolidating their knowledge into a knowledge structure.

Knowledge structures consist of the problem space that explains the mechanism of how all the variables work together to manifest themselves as the symptom (‘problem’) [19, 52]. Knowledge structures describe the degree to which an individual organizes information elements in memory. These include an individual’s understanding of the facts, concepts, and their relationships embedded within the problem space. Theorists argue that strong knowledge structures facilitate subsequent learning when new information is presented [5, 52]. Furthermore, it is posited that retrieval is impacted by the construction of the knowledge structure; that is, a well-constructed knowledge structure facilitates efficient pathways when learners need to reference an idea. It is hypothesized that learning can be thought of as the degree to which learners alter their knowledge structures. As it relates to education, theorists contend that the contextualized nature and emphasis on problem-solving in inquiry-based learning strategies facilitate the development of robust knowledge structures.

Proponents of inquiry-based learning suggest that the emphasis on ill-structured problems better bridges the differences that exist between knowledge structures of experts and novices [46, 68]. Whereas the knowledge structures of novices is characterized by misconceptions, disconnections, and surface level understanding of the problem space, experts include a more complete, structural-level understanding. Experts’ knowledge structures are often defined as a more holistic, schematic organization of information, whereby the concepts in the problem space are organized in a relational and semantic manner [58]. In turn, this allows experts to approach problem-solving in a decentralized way that supports causal reasoning [45]. In contrast to experts, studies show that

novices' knowledge structures often focus on readily available and most salient concepts, while tending to overlook more foundational concepts that are not as obvious [20, 48]. In addition, novices' explanations are linear and focus on a single cause [37, 123]

### **4.2.2 Assessment in ill-structured problem solving**

An important part of the problem solving process is the ability to understand the most relevant elements of the problem space. Given that networks can be depicted through nodes and links [52], concept and causal maps are becoming increasingly popular forms of assessment [90]. Clariana [11] suggests there are four unique aspects relating to mapping of the problem space. First, the open-ended nature of ill-structured problems requires learners to go through the process of recalling and selecting which concepts (i.e., nodes) to include in a map. Upon completion, students must be able to engage in meaning-making about how their concepts are related. Doing so generates important insights about the structure of causations (when a directed link goes from one concept to another) and associations (when a link is undirected or two directed links are reciprocal). Causal maps also elucidate what learners choose to select from the problem space (extent of knowledge): the distance of those relationships (proximity), the links between the ideas (lexical association), and the perceived final state of the problem representation (conditional knowledge) [12].

Although concept maps may serve to understand a student's knowledge construction, the ill-structured nature of problem solving and design of concept maps creates a significant challenge for educators. Indeed, research finds that educators cite assessment as a significant barrier, which precludes their proclivity to implement inquiry-based learning and ill-structured problem solving in the classroom [122, 133].

Challenges particularly arise at two steps of the assessment. First, variations in language (e.g., 'cardiac arrest', 'heart attack') can create an important confound that makes assessment inefficient

and impractical. This problem may be prevented from appearing in the first place when students are limited to using a complete list of concepts from the problem space, which may include ‘distractors’ or ‘misleading’ concepts [104]. For instance, in participatory modeling studies, linguistic variability is limited by using a set of terms standardized through independent focus groups [35]. However, imposing such limitations may produce two maps that look more similar than the individuals’ knowledge structures [81]. The alternative is to place no restriction on the use of concept names, and then to go through an alignment phase in which educators identify equivalent terms either manually [32] or through algorithms such as recommender systems [38].

Second, once variations in language have been resolved, educators have to compare maps structurally. Early approaches have focused on counting elements that are present in both the student and expert map, or only in the expert map. For instance, such approaches can point out which links are shared, and which links the student may miss [126]. More recent approaches have treated causal maps as graphs, using structural metrics derived from graph theory, social network analysis, or network science. For instance, the diameter of the spanning tree was proven to be a reliable measure [52]. More recently, Lavin and colleagues demonstrated using 264 maps that specific types of centrality indices (i.e., metrics to score the importance of nodes) could be used to infer that individuals or groups would make similar predictions [81]. Other approaches and experimental studies using structural matching, semantic matching, overlap measures, and propositional matching have been detailed by Krabbe [80]. In sum, assessment approaches mostly employ tools from graph theory designed to measure structures in one map (e.g., diameter of the spanning tree), and then two maps were compared with respect to their individual measures. This is an indirect approach to comparison, using tools that were not specifically designed for this purpose. In contrast, graph theory (and particularly as it relates to machine learning) possesses many tools to specifically compare maps. That is, tools exist that can take in two maps, and compute the distance between these maps. The next section seeks to address the paucity of graph comparison methods in assessment by providing a brief overview of these tools and their possible uses in educational research.

## 4.3 Using graph comparison methods for assessment

### 4.3.1 An intuitive introduction to graph comparison

There are three broad approaches to comparing a student map with an expert map. While they are grounded in graph theory and machine learning, this section provides an intuitive overview of these approaches. Details and pointers to formal specifications are provided in the following sections. Examples from assessment are reinforced with the metaphor of comparing houses.

First, we can measure the number of changes necessary to transform one structure into another. Graph Edit Distance (GED) accomplishes it by finding an efficient sequence of transformation. In the case of houses, we can compare their plans side by side, and see what changes are necessary to turn one house into the other, such as adding a bedroom, removing a bathroom, or re-labeling a bedroom as a home office. In the case of maps, we can add or remove nodes and edges, or re-label a node's name. The output of the GED is a single number (i.e., the distance), but the process to compute it is also of interest for assessment.

Second, rather than doing a possibly long sequence of minute changes, we can ask whether 'the big picture' is similar in two structures. The idea of a graph kernels is to focus on the core of a structure. For instance, the core of a house could be the size of its rooms. We would thus compute the distribution of room sizes in both houses, and compute the distance between these discrete distributions. Similarly to GED, the output of a kernel is a single number, but it measures the discrepancy between distributions of user-defined features. For example, taking the approach of Kotovsky and colleagues [79], we could measure the distribution of the number of branches per node, as a proxy to a map's 'complexity'.

Third, we could apply a set of metrics to the structures, and compare them based on the results for each individual metric. In the case of houses, one may measure taxes and surface area, and then

each house can be plotted into a 2D space where taxes are on the x-axis and surface area on the y-axis. The final result of a graph embedding is thus a single number comparing points in space. For a map, we can measure N features, and plot the map in a space of N dimensions.

Within each approach, we have to state precisely what we value. For instance, in graph edit distance, is it worst when a student has a link that the expert does not have, or misses a link that the expert does have? In graph kernels, which structures are indicative of learning? For graph embedding, which features should we extract from a map? Finally, we need to select an algorithm that efficiently accomplishes the computations. There are dozens of algorithms to compute the graph edit distance, numerous ways to compare two distributions, and several methods to compare two points in space. The remainder of this section details each approach, including its implications for the assessment of digital knowledge maps in education, and how to choose values as well as algorithms in this context.

### **4.3.2 Graph Edit Distance (GED)**

There are two broad approaches to graph matching. On the one hand, we can require an exact match, when asking questions such as ‘are the student and expert maps identical’, or ‘what is the largest part of the expert map that the student got right’. However, these questions may not be of practical relevance (e.g., the student and expert maps are very unlikely to be identical), and the answer may support summative assessment more than formative assessment. In addition, these questions are respectively known as graph isomorphism and subgraph isomorphism, and solving them in a reasonable amount of time remains an active area of research [4, 9]. For all these reasons, solutions such as Graph Edit Distance (GED) have been developed to handle an inexact match, in which differences between maps are penalized rather than forbidden.

In Graph Edit Distance, we look for an edit path, that is, a sequence of operations that transforms the student map (source) into the expert map (target). Operations can be performed on edges (deletion, insertion) and nodes (deletion, insertion, re-labeling). It is always possible to transform a map into another one: for instance, we could delete all of the students' nodes and edges, and then add all of the expert's nodes and edges. We are thus interested in finding the best path. When all operations are viewed as equally important by the instructor, then the best path minimizes the number of operations.

As the method is flexible, instructors can also state that some operations reveal more of a misunderstanding from the student than others. For example, studies suggest that students struggle to remove extraneous concepts from a problem space [45]. In addition, studies find that students focus on the surface level characteristics rather than the less-salient concepts that are more central to the problem [20, 58]. In these cases, the best path is one that minimizes the sum of the operations' costs. The costs of operations such as adding/removing nodes is usually set to a positive constant, and the same applies to edges if they do not have a label. The costs for re-labeling have often been approached from a theoretical perspective [115], where labels are seen as vectors of characters and pair-wise differences are computed between characters (e.g., 'bad' and 'sad' differ by only 1, whereas 'bad' and 'not good' differ on all 8 characters). In contrast, for assessment, we are interested in the semantics of the words. Methods exist to automatically measure the strength of association between terms [110], but they have not yet been widely applied in the context of comparing maps, perhaps due to the paucity of use for graph comparison techniques in educational research. Our recommendation would be to resolve variations in language before computing the GED. In other words, a preprocessing step would align terms and that would avoid penalizing for variations in language during the GED. All remaining differences in labels would incur the same constant cost.

Once the instructor has preprocessed the maps (to resolve variations in language) and identified suitable costs, then the GED can be computed. This is useful for summative assessment (as the

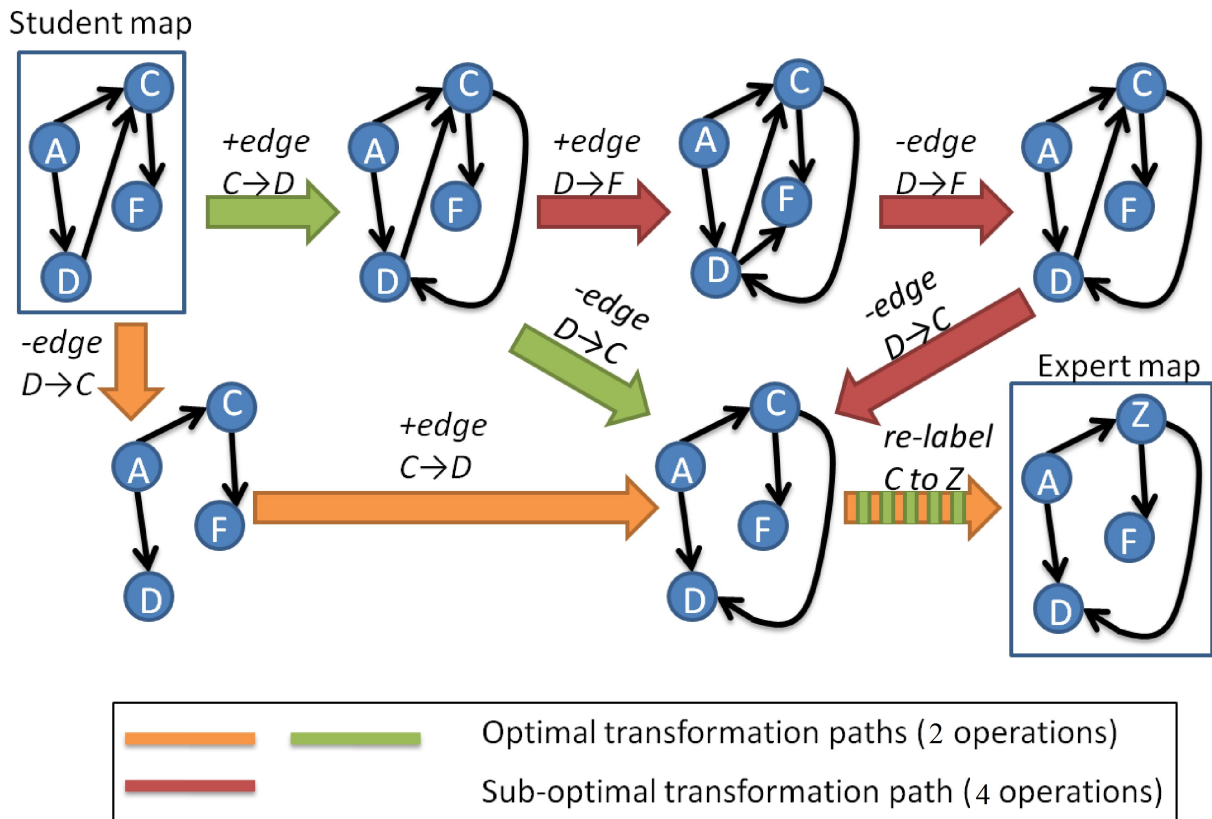


Figure 4.2: A student map (top left) can be transformed into an expert map (bottom right) through sequences of operations, some of which are equally short (green, orange) and some of which go through unnecessary steps (red).



GED is a number summarizing how ‘close’ the student got to the expert), and particularly for formative assessment as computing the GED shows how to transform the student map into the expert’s. This is related to the concept of action sequence in educational research. In contrast to general guidelines or ‘one size fits all’ approaches to identifying general action sequences leading to accurate maps [61], computing the GED can create an entirely personalized action sequence for a given student’s map. Future research may explore whether these personalized action sequences do cluster across students, and if so, based on which individual characteristics. Clustering and sequential pattern mining would provide powerful methods towards this objective [92].

### **4.3.3 Graph Kernels**

A graph kernel considers a graph as being made of an unordered collection of simpler patterns. The matching problem thus consists of extracting and comparing these patterns. The patterns are user-defined, thus they vary depending on the emphasis and context of each study. These techniques are particularly used to compare biological networks, and they are summarized in this context by Mueller et al. [88]. Patterns have included ‘graphlets’ (i.e., all subgraphs with 3, 4, or 5 nodes) [113], which are similar to the ideas of motifs and the Triad Significance Profile used in social networks and regulatory networks for biology [73, 105], trees, when comparing the structure of molecules [85], and cycles, which have also been used for molecules. When it relates to assessing causal maps, patterns can be used to examine the level of systems thinking exhibited by a student. In the classification of systems, independent nodes are at the lowest level of systems thinking, while edges can be slightly higher [86]. When going even higher, we start looking at feedback loops, also known as cycles. A set of nodes are in a loop if, starting from any node in the set, we can follow a sequence of edges that ends at this node. Loops capture a student’s understanding that ‘a change can be initiated everywhere in an event circle and after a certain

time be read off as either cause or effect elsewhere in a system' [114]. Studies have shown that loops were absent from many causal maps about a variety of problems even when they drive the dynamics of these problems in the real world [6]. It is thus important when comparing the maps made by individuals to look at their cycles. That is, a relevant kernel for assessment would be the distribution of cycles, which counts the number of cycles (y-axis) of each length (x-axis). In sum, comparing two maps would compare their distributions of cycles.

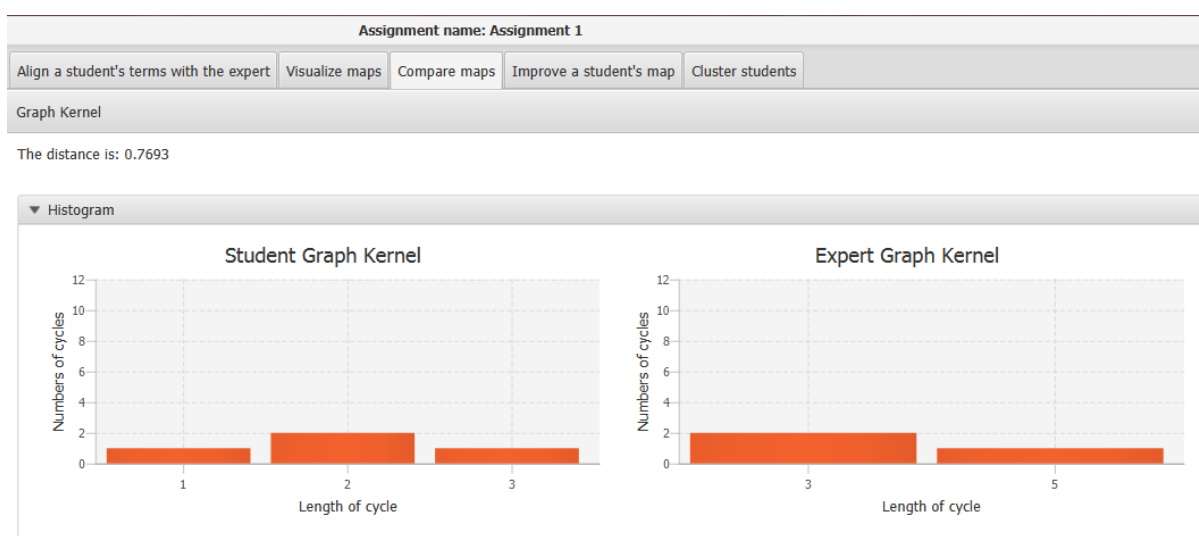


Figure 4.3: Comparison of two maps based on cycle distribution.

There are several important differences with using the GED described in the previous section. The GED takes a very detailed view of all the operations needed for the transformation: it measures what is different, and tells the student how to fix it. This serves as a form of knowledgeable peer and scaffolding as students are alerted to gaps in understanding. When students become aware of their knowledge deficiencies, they are encouraged to reflect and later iterate their problem solving [25, 50, 125]. The kernels would not give us an action sequence, but they would tell us the bigger picture of how a student ‘thinks’ compared to the expert. For instance, the GED may tell the student to add 4 links because the expert has them. However, three of those links may be used

to close a loop, and it is thus particularly important that they are present, whereas the fourth one only connects to a peripheral concept.

An action sequence could still be derived from kernels, for instance by listing all the loops that the students missed, and highlighting which specific edges have to be added for these loops. Similarly, we can list loops that the students claimed but the expert does not endorse. An experimental study may compare whether this action sequence or the one generated by GED yields either more accurate maps, or a better understanding of the problem.

#### **4.3.4 Graph Embeddings**

Several measures from graph theory have been mentioned in relation to the assessment of causal maps [80], including the diameter of the spanning tree, the number of components (i.e., disconnected parts of a map), or the density (i.e., ratio of edges present to the total number of edges that could connect the nodes). Each one of these measures produces one number. Comparing two maps by looking at each individual measure can be a challenge. For instance, are two maps similar if both have a single component, but one is denser than the other? Would they be more similar if one was less dense, but had a longer spanning tree? A composite score can be created to provide a single number based on a set of metrics. While this can be achieved by approaches such as taking the weighted sum of the underlying metrics, it raises the question of how to set appropriate weights for each metric, and whether we should account for interactions between metrics to avoid double- or triple-counting. In contrast, graph embeddings provide a mathematical framework. The idea is to ‘embed’ an object (i.e., a map) into a vector space, where the distance between the embedded objects serves as a proxy for the actual distance between the objects themselves [44].

In the case of embedding causal maps for assessment, the instructor decides on  $N$  metrics to use. Each map is then transformed into a vector with  $N$  coordinates, whose values are the graph’s

scores on each selected metric. For instance, Figure 1.6 shows how three maps can be positioned in a 3-dimensional space based on their number of nodes, number of edges, and average number of edges per node (i.e., average degree). Using only 2 or 3 metrics allows instructors to conveniently see each map as a point in space. When using 4 or more metrics, results can still be visualized but the multi-dimensionality requires the use of techniques such as parallel coordinates [57], which may be less intuitive. Once the maps have been transformed into vectors based on selected metrics, the vectors can be compared. This can inform instructors on whether a student is heading in the same direction as the expert. For instance, a student may have several nodes, many edges, and a few cycles. If the expert has these elements in similar proportions, but in a larger map, then we suppose that the student's structure is going in the right way and may need time to mature. Conversely, if the student has few edges and no cycles, then an intervention may be needed to set the student on the path to success.

## 4.4 Implementation

Our implementation grew out of the Incremental Thesaurus for Assessing Causal Maps (ITACM) software. The software was first released to assist with reducing linguistic variability when assessing maps [32], and the comparison then consisted of counting the percentage of factors whose names were equivalent in two maps. The software has three broad limitations. First, it did not allow for the comprehensive forms of comparison summarized in section 1.2.3. Second, it was a desktop application, although previous studies have 'advised to use software with a web interface or client-server architecture that allows to retrieve a concept map from different work places through the Internet' [80]. Third, the reduction of linguistic variability was a labor intensive process, where the computer was only able to recognize whether two terms were set as equivalent by the instructor previously.

The second release, ITACMv2 [38], addressed the last two limitations. Taking a client/server architecture allows the software to support a community of practice. Given that educators struggle with the initial preparation of inquiry-based learning, allowing users to share maps with their peers allows us to save time in terms of onboarding [15, 122]. Rather than starting with a ‘blank slate’ to redesign the curriculum, sharing resources using an open education format allows educators to leverage expertise within their own learning communities, which allows dissemination of best practices within peer networks. The use of recommender systems also allowed instructors to identify potential equivalences between terms used by the student and the expert, which results in a faster alignment process.

Our newest release, ITACMv3, addresses the last limitation by giving access to all three approaches to comparison presented in this chapter. The implementation of the Graph Edit Distance uses the beam search approach [98, 99]. The intuition is as follows. Given a student map with  $N$  nodes and an expert map with  $M$  nodes, we can start by taking one of the student’s concepts and see onto which expert’s concept to map it. There are  $M$  possibilities. We can continue the process with the next concept from the student, and there are still  $M$  possibilities as two concepts from the student may refer to the same concept in the expert. There are thus on the order of  $M^N$  edit paths. Finding the right one is similar to a game of chess, in which a very large tree is created to compute each possible move, the resulting board configuration, and each possible move within each configuration. Tree-search based methods such as the A\* algorithm thus estimate the cost of each of the possible branches. Beam search is an improvement introduced by Riesen that further prunes the search tree. Five other alternative implementations are presented in section 4.2 of [96].

Our implementation for graph kernels uses cycles. Cycles are listed using Johnson’s algorithm [63]. The problem of comparing two maps has been transformed into comparing the distribution of their cycles. A statistical approach to measuring differences between distributions is to use an f-divergence, which is a type of function. Specific functions include the Hellinger distance and the Kullback-Leibler divergence (known as KL-divergence). KL-divergence is widely used

in information retrieval and machine learning. But we implemented Hellinger distance, as KL-divergence requires one map to be derived from another map and it is possible that a student might totally differ from the expert's map. A Hellinger distance of 0 suggests we can expect a similar behavior, while 1 suggests that the distributions behave very differently.

Finally, for graph embeddings, we used three metrics: the number of nodes, number of edges, and the graph density. The similarity between two maps is computed using the cosine similarity between their corresponding vectors. Two vectors with the same orientation have a cosine similarity of 1, intuitively meaning that the two maps are 'thinking in the same direction'. Theoretically, diametrically opposed vectors have a cosine similarity of -1, but this situation cannot exist here since the metrics produce strictly positive numbers. Thus, the minimum cosine similarity is 0, when vectors are orthogonal. Our interface is shown in Figure 4.4. We recommend that instructors first align the terms to reduce linguistic variability (first tab), optionally inspect the maps visually (second tab), and then start to compare maps (third tab). Explanations are provided for each metric to aid with interpretability of results.

## 4.5 Discussion

Ill-structured problems serve as the foundation of inquiry-based learning. These problems are defined as having multiple problem elements, unclear goals, and constraints. An individual must thus articulate multiple solution paths and criteria for the proposed resolution [69]. In doing so, educators are able to go beyond the traditional, didactic forms of learning as they pursue higher order learning outcomes [42, 45]. Indeed, various studies cite the benefits of ill-structured problem solving afforded by inquiry-based learning when properly supported [76, 82, 131].

The aforementioned instructional strategies have given rise to alternative forms of knowledge representation. Hung [51] contends that students should be evaluated on their ability to 'articulate

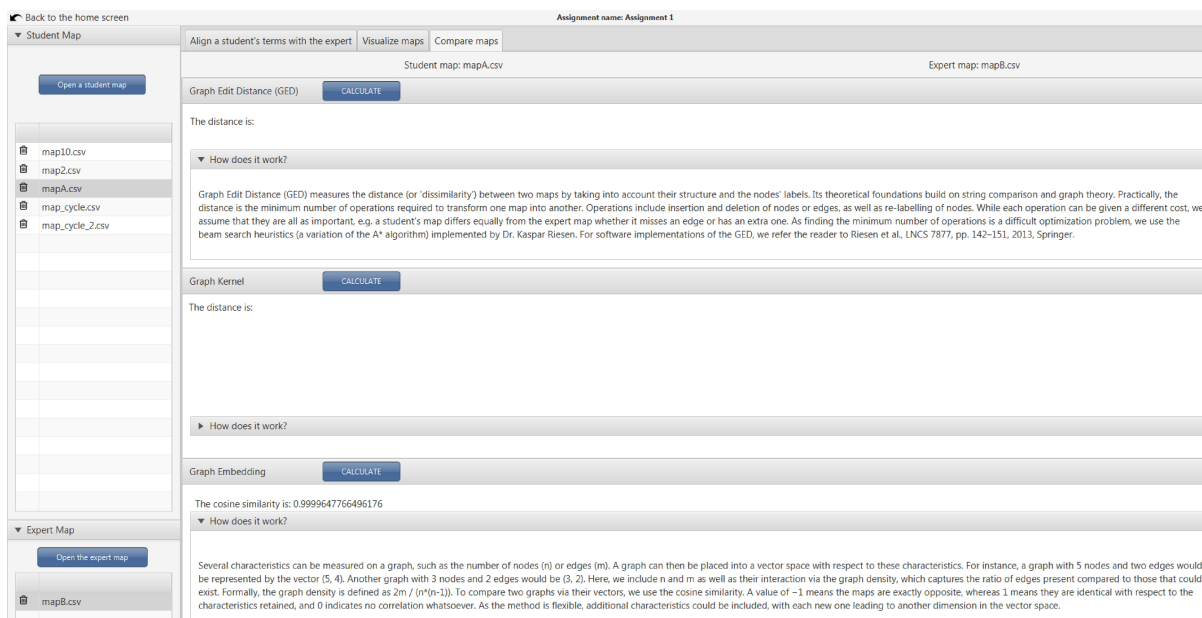


Figure 4.4: The newest version of the ITACM software includes a ‘Compare maps’ tab, in which educators have access to an implementation of Graph Edit Distance, Graph Kernel, and Graph Embedding. Details of the specific implementation can be accessed by expanding the panel ‘How does it work?’ within a metric.

the critical elements of the problem, their process for solving it, and the solution proposed and defend their proposed solution and the rationale, rather than whether they match predetermined answers’ (p. 547). One evidence of how learners solve ill-structured problems is through causal maps, which require learners to articulate relationships consisting of the concepts’ antecedents and consequents [69, 132]. Causal maps are especially beneficial because they provide insight into the connections that learners make between elements of the problem space, along with affording opportunities to depict multiple solution paths for ill-structured problems [12, 69].

Despite the purported benefits of inquiry-based learning, educators face unique challenges in implementation [51, 133]. One emergent confound for educators is how to assess ill-structured problems accurately and efficiently. In terms of the former, ill-structured problems are defined by their ambiguity. Moreover, ambiguity of assessment is impractical from a time-management perspective. Research suggests that educators cite assessment as a primary reason for failing to

sustain inquiry-based learning in the classroom [15, 122]. To date, there have been a number of attempts to resolve the assessment challenge in inquiry-based learning. One approach is ‘referent-free’, as the causal map of each student is assessed independently with respect to certain desired structures such as the total number of links [61]. The other approach assesses a student map using an expert map as referent. This is the approach studied in this chapter. Experimental studies have demonstrated that using a referent can foster a better understanding of the problem [54, 127], but a manual comparison of maps is too time-consuming and calls for the development of digital tools [126]. Several such tools have been developed, but they are often limited in contrasting low-level features (e.g., number of links that a student has or misses vis-a-vis the expert) or a set of properties (e.g., number of disconnected components within the maps). In contrast, graph theory and machine learning have created many algorithms specialized for comparing maps [24, 129]. This chapter has introduced three categories of algorithms for map comparison, with a focus on their applicability for the assessment of causal maps.

We showed that Graph Edit Distance (GED) provides both an estimation of the difference between maps (for summative assessment) and an action sequence personalized to assist a specific student in bridging the gap with the expert map (for formative assessment). An open question is whether the personalized action sequences tend to cluster, and for which student characteristics. This may assist instructors in better analyzing the learning journeys of a group of students or, similar to the design of group-level interventions in health [30], it could assist with the identification of more homogeneous groups of students. An open technical challenge is to estimate the semantic relatedness of terms used by students and the expert, such that the action sequence can deliver better explanations than merely asking students to change a concept’s name. We discussed the benefits of graph kernels for assessing systems thinking in students, by extracting the distribution of feedback loops in their maps, and comparing it with the expert’s. An open challenge is to use the differences of loops between the student and the expert to provide an action sequence. A possible benefit is



a higher-level view on why the student needs to consider certain edges (e.g., because they close a loop and give rise to certain dynamics), which the GED does not provide.

Lastly, we examined the potential of graph embeddings to create a composite score and reveal whether a student is thinking in the same way as the expert. This may be most useful to provide feedback as the student gradually develops the map. An open question would be to determine the threshold of the difference between the student's map and the expert's, such that timely recommendations can be made to assist the student in staying on the right path. We acknowledge that taking such approaches to assessing causal maps requires a leap forward, given that current software in educational research implements none of these approaches. We have thus provided one implementation (ITACMv3) for all three approaches, which educators can use within their own learning communities. Experimentally evaluating and contrasting these three approaches are long-term objectives that are well-worth addressing for the research community.

# CHAPTER 5

## DIGITAL TOOL TO IMPROVE STUDENT'S MAP BY PROVIDING THEM WITH FEEDBACK

### 5.1 Introduction

The existing research in structural assessment of knowledge (SAK) [128] primarily focuses on conceptual maps in various forms. However, there is precedent for addressing ill-structured problems by modeling them as causal maps [116]. Since causal maps can be considered as a more highly constrained form of concept maps, many of the same techniques used for evaluating and comparing concept maps can be applied to causal maps as well. Since the relationships available in causal maps are highly constrained (only directed edges representing positive and negative causal influence are allowed), comparing the relationships between concepts is relatively simple, without any need to determine how closely the specified relationships in two maps match. In addition, causal maps are widely used in certain areas already (e.g., modeling and simulation, system dynamics), so there exists a community of practitioners already familiar with their use.

Wu et al. point out that new information should be taught in relation to the learner's original

This chapter is a joint contribution of:

- **Vishrant K. Gupta**, Russell Lankenau, David Offord and Adam Wykle.

My contributions consisted of developing the software, implementing and testing it by leading the team in discussions to provide technical solutions for generating feedback.

knowledge structures [134]. This suggests to the authors that whenever possible, the terms used by students creating concept maps should be preserved. It is proposed that this increases usability of the feedback produced by the software as well as contextualizing suggestions within the learner's existing knowledge. Although it is recognized that maps that are created without restricting terms are more useful to students [49], the prevailing attitude is that comparison of maps using terms are extremely difficult to compare and evaluate [49, 87, 106, 134]. The software uses a subject-area specific thesaurus database constructed by users to allow terms in one concept map to be aligned with terms used in an expert map to facilitate comparison. This feature allows concept maps to be effectively analyzed while still maintaining the terms specified by the student who built the original map.

The area of focus in applying causal maps to SAK is what Trumpower and Vanapalli refer to as "SAK for Learning", i.e., the use of SAK to provide feedback for students and guide the learning process [87]. Wu et al. developed a technique for generating meaningful feedback for students by providing hints regarding changes that should be made to their maps in order to bring them closer to those provided by experts [134]. These hints were in the form of "there is a missing notion related to Concept A" and "there is a missing connection related to Concept A and some other Concept" [134]. Students would submit their map, make changes based on the provided hints, and then submit their revised map. They would also be provided a score to give them an idea of the overall correctness of their map. The system also provided links to supplementary information that would provide direction surrounding implementation of each hint [134]. In the study conducted by Wu et al. where the same supplementary information was available to students using a system that provided feedback and a control group, students using the feedback generation system accessed the information more frequently [134]. While useful, such feedback must typically be created by instructors, and may be a deterrent to adoption of such a system.

In addition to the individual relationships represented in concept maps, causal maps are typically analyzed for the presence of large system-level structures that can cause large system-wide

changes. Two such structures are the presence of disjoint paths between concepts and the presence of loops. Both disjoint paths and loops can exert a large amount of influence on the behavior of a system, so they are important features for students to identify in their own maps and those of experts.

The software provides suggestions for improving the student's map based on direct comparison between student and expert map. Also it indicates whether the conception of a student about the system as a whole is lacking or not. In addition to direct suggestions about adding and removing a node it is necessary to identify higher-level structures that can be used to guide the student in developing their systems thinking skills. Two such structures in causal maps are disjoint paths and loops.

## **5.2 Disjoint paths and loops**

Disjoint paths in a causal map represent multiple ways in which one factor can influence another, and can help us to identify areas of the model where multiple influences combine to reinforce or balance system-level effects. Similarly to causal loops, if we have an odd number of edges with negative causation in a path, the net effect of the path will be negative, otherwise it will be positive. Identifying areas where students have missed disjoint paths is important because people tend to miss system-scale structures when building models due to our inability to consider all factors involved simultaneously. Knowing that additional disjoint paths exist between two nodes can provide the student an area of focus, allowing them to concentrate on specific areas during the learning process. In addition, the purpose in modeling is often to determine areas where interventions can be made. Disjoint paths provide useful information in making this kind of determination: a node where many disjoint paths originate could be a powerful lever for changing the node where those paths terminate, but if these paths have mixed positive and negative net effects, the impact of an in-

tervention might be difficult to predict. In either case, without identifying the set of disjoint paths, we can't decide either way.

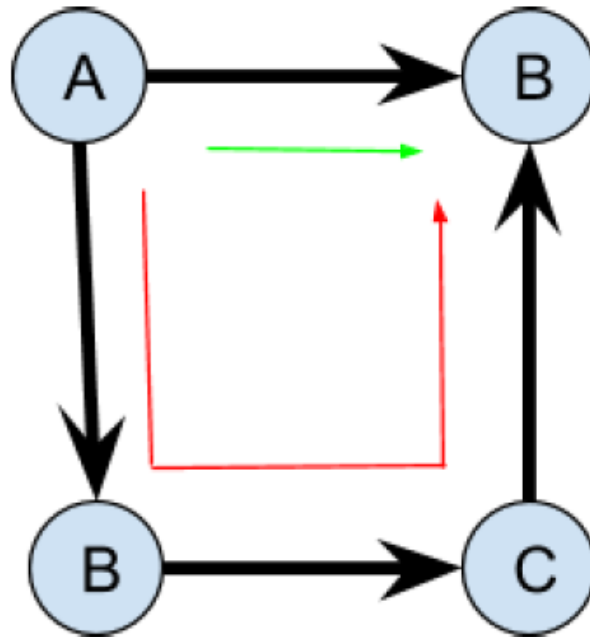


Figure 5.1: Example: Disjoint path.

In order to identify multiple disjoint paths, we must examine each pair of nodes in the map independently. For each pair of nodes, we can determine the set of disjoint paths between them by using Tarjan's extension of Suurballe's algorithm, which discovers all paths between two nodes in a graph that do not contain any shared edges [121]. Each iteration of the algorithm requires running Dijkstra's algorithm, followed by a modification to the set of edges based on the edges contained in the shortest path. Each iteration discovers a single path, in order of length, so it is fairly easy to restrict our search to the most direct set of disjoint paths if we want to reduce the time required to analyze a map.

As humans we tend to reduce the complexity of concepts as a way to prevent information overload and to reduce the amount of mental energy required for a task. As a result, people usually think acyclically, which leads to ignoring feedback loops between concepts. In order to help the

student see past their own biases, the system allows instructors to detect loops that may be missing from the student's graph.

A loop in a casual map can be one of two kinds, a reinforcing loop or a balancing loop. Reinforcing loops are cycles that grow or collapse a system over the course of the simulation. If we have an odd number of edges with positive or negative causation in a loop, the net effect will not be zero, which makes it a reinforcing loop. Balancing loops, on the other hand, will continue to stabilize the system over the life of the simulation by maintaining the stock of variables. Both of these are very important because they can stabilize or destabilize a model very quickly. By inserting even a single missing loop, a student's map could fundamentally change. This could have drastic effects on the way the simulation runs and change the outcome in major ways.

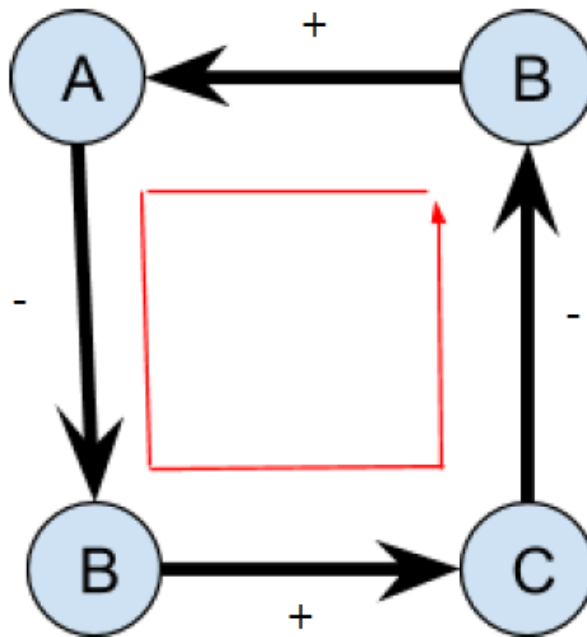


Figure 5.2: Example: A balancing loop.

### 5.2.1 Implementation of disjoint path and loops

To implement the insertion of missing loops into the student's graph we first need to be able to recognize loops in the instructor's map. In this software Johnson's algorithm was used to identify loops [64]. After we have the list of loops in the expert's map and the student's map, we can use them to analyze the student's map and identify what the map is missing. Once the missing components have been identified, we can now insert the required nodes for the loop to exist. This would be an inexpensive way to implement the discovery of loops and insertion of the nodes required to create them.

The goal is to add a new feature to allow an instructor to quickly determine how a student's causal map, created with no restrictions on available concept terms, differs from the expert map, which may have been created using a different but analogous set of terms, and to create a set of feedback items that can be used to improve the student's map. Our focus was on causal maps, where system-level interactions are extremely important, so it was important to generate feedback that did not simply rely on a direct comparison of nodes and links.

Wu et al. implemented a system for generating immediate feedback for students using concept maps with a predefined set of terms and relationships, and a set of weighting factors corresponding to the relative importance of the relationships. The process of annotating relationships with weighing factors is an additional task required of the expert creating the map. This factor was only used to calculate the "correctness" score of the student map [134]. Rather than requiring any additional annotation of the map in order to provide meaningful feedback, we focused on system-level features. This was only possible because causal maps exhibit system-level behavior indicated by multiple-hop paths. Since more general forms of concept map allow for many different types of relationships, only single-hop paths are typically examined.

In order to provide feedback on situations where the student failed to identify that there were multiple paths of influence between two nodes, we implemented disjoint path detection, using both an edge-disjoint path detection algorithm and a node-disjoint path detection algorithm. The node-disjoint path detection algorithm was based on a transformation scheme outlined in Suurballe's 1974 paper, but used the same underlying technique as the edge-disjoint version [120]. We consider detection of disjoint paths important because it can identify situations where a student only identified the most obvious way in which one concept influences another and did not consider alternatives.

The other type of system-level structure we identified were loops. Loops represent feedback within a system, either amplifying or damping down different effects. Many complex systems contain loops, so the absence of loops is another indication that the student may not have considered all possible influences present in the system.

Rather than simply displaying the feedback directly in the tool, we also provided a method for exporting the feedback as a CSV file suitable for analysis using standard spreadsheet tools or custom analysis software. By simplifying the process of getting data out of the tool, we hope to make it easier to use for future analysis.

### **5.3 Design Process**

In order to provide a tool that allows instructors to easily evaluate causal maps developed by students and guide students in making improvements to their maps, we began by identifying the intended users of the software and considering possible user interface designs with those users in mind. Although students submit maps to be evaluated, the users of the software are instructors.



The expert maps used for comparison may or may not have been created by the instructor using the software. We assumed that the users were familiar with concept maps in general.

Our design goals for the new functionality were to provide the instructor with feedback that could be conveyed to the student at their discretion in order to make the student's map more closely resemble the expert's map, and to use a simple user interface that would allow the instructor to orient him or herself within the student's map. Another design guideline we established early on was that we would use the phrasing specified by the student whenever possible, so that the student could more easily contextualize the changes that were required. Ho, Kumar, and Velan point out that "meaningful learning" involves "integration of new knowledge with existing understanding" [49], which lends support to this approach.

With our design goals laid out, we began brainstorming different approaches. We decided fairly early on that we would not require instructors to annotate the expert maps with justifications explaining the reasoning for including specific structures. This was deemed to be too much additional work for the instructor. After some discussion of possible approaches, we categorized our possible approaches according to two distinct axes: the level of detail included in the feedback, and the style of feedback provided. We considered both textual and graphical displays in terms of feedback style, and various levels of detail ranging from a step-by-step modification plan for the student map to a very general display showing the merged maps with differences highlighted, similarly to the approach used by Sanwar and Trumpower [106]. After building wireframes of each of the possible user interfaces, we selected a combination of textual and graphical approaches to provide both specific feedback and context within the student map (see Figure 5.3). Rather than provide incremental feedback (e.g., the one or two feedback items judged most important), we decided to provide the entire set of feedback items in a list. This allows the instructor to judge which feedback items most effectively reinforce the learning outcomes they are working towards.

We implemented the desired features in several phases in order to gather feedback on the usability of the tool. We began by only suggesting changes to a student map based on differences in

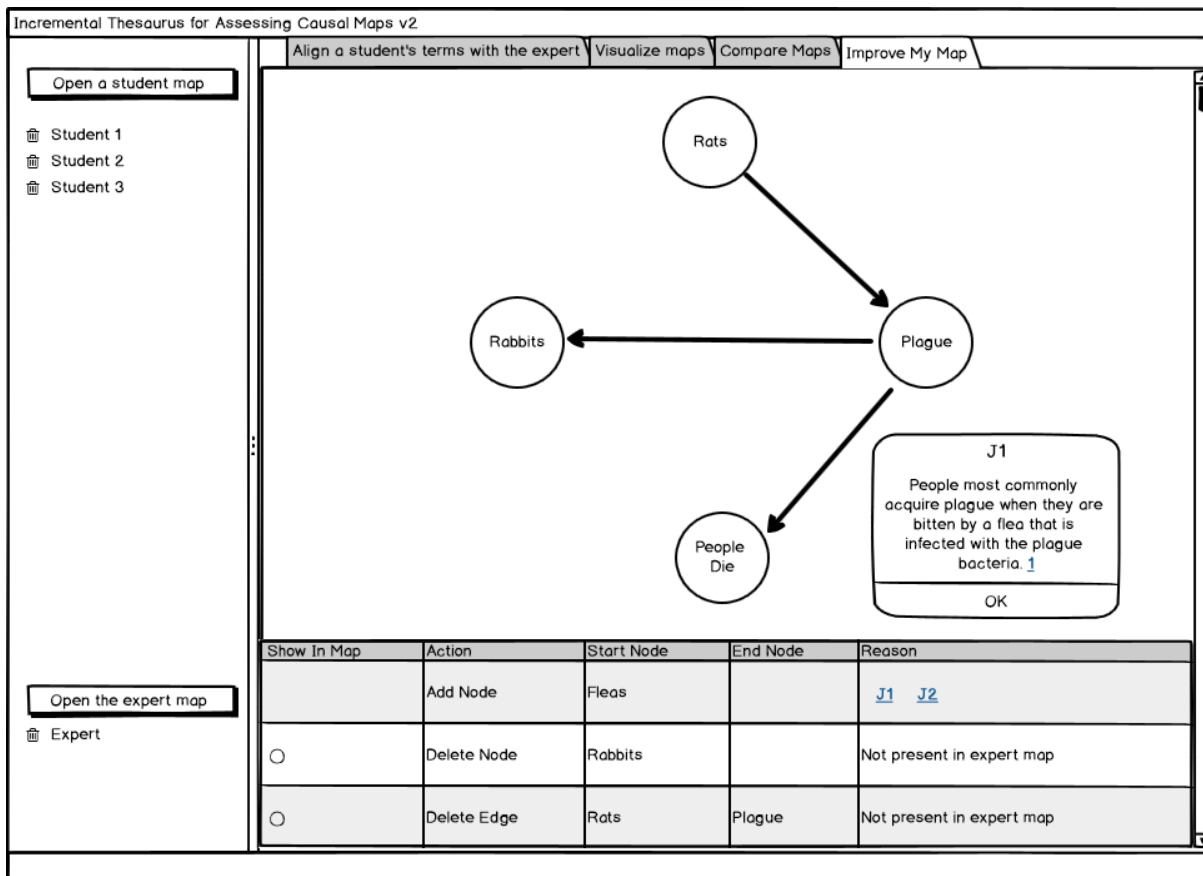


Figure 5.3: Wireframe of the proposed solution to provide feedback.

the set of nodes or edges between the student map and the expert map. We then moved on to the detection of edges that were part of loops or disjoint paths between concepts present in the expert's map but missing from the student's. Each feedback item was justified with a set of reasons listed in the table, such as “The node ‘concept 1’ was present in the expert map” or “The edge between ‘concept 1’ and ‘concept 2’ would complete a loop present in the expert map.” When multiple reasons were applicable, we listed all possible reasons. In addition to iterating on the types of features detected, we also simplified the user interface at the same time, removing radio buttons which were not necessary, and selecting a single color to highlight concepts and relationships which were mentioned by the feedback item selected by the user. We also added the ability to export the full list of feedback items to a CSV file (see Figure 5.4).

Action	Start Node	End Node	Reason(s)
Add edge	a	a	R1 R2

Figure 5.4: An example of how feedback is provided.

## CHAPTER 6

### CONCLUSION

Well-structured problems are the most common type of classroom exercise, but they do not capture the complex context of the real world. Ill-structured problems are much more representative of problems encountered in the real world, but their solutions are much more difficult to access. Ill-structured problems often require context outside the problem statement and have a complex solution space with multiple valid solutions.

Causal networks are a powerful tool for representing the way individuals conceptualize solutions to complex problems. By documenting the factors involved in a problem and their relationship to other factors (as shown in Figure 6.1), we can analyze the way that a system reacts when certain factors are changed. A causal map thus can be used by both students and teachers to visually represent their knowledge and understanding of a system. These maps can be used in all types of classroom situations as described by Spector et al. for MOOCs [117] and M. Wijnen et al. for a classroom-based course [122, 133]. The visual representation of a system using a causal map promotes cause-and-effect thinking of a system and thus helps in understanding it in a better way. A causal map can be a way to organize information about the system in a meaningful and logical way that shows a relationship between different concepts using links between nodes.

A causal map can be used to solve ill-structured problems (ISP) faced by practitioners as a result of a broader set of solutions produced against an open-ended question asked in a classroom environment that are often well-structured as they have predefined answers. The solutions produced by a student in the form of a causal map are often contextual [67]. A student causal map can be contrasted with an expert's map, but a perfect comparison is a graph isomorphism problem which is NP-complete and requires exponential computation time.

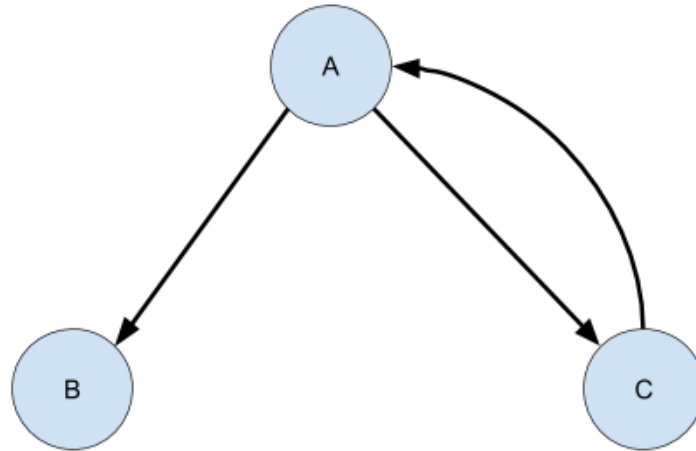


Figure 6.1: Demonstration of a factor influencing another factor.

## 6.1 Contribution

Causal networks built by students can demonstrate their understanding of a problem but it is difficult and time-consuming to access their submissions and evaluate their correctness. Comparing student networks to expert networks is the best way to accomplish this. When comparing two causal networks, we are faced with three problems: determining how the included concepts compare between the two networks, calculating how similar they are, and providing feedback for the student on how to improve.

Each individual creating a causal network will have slight variations (as illustrated in Figure 6.2) in the way they name the included concepts and the level of abstraction they use. Before we can do any meaningful comparison between two networks, we must determine how the terms used in the networks match up. Performing this network comparison automatically is extremely

time-consuming because the computer must analyze every possible mapping between concepts as shown in Figure 6.3.

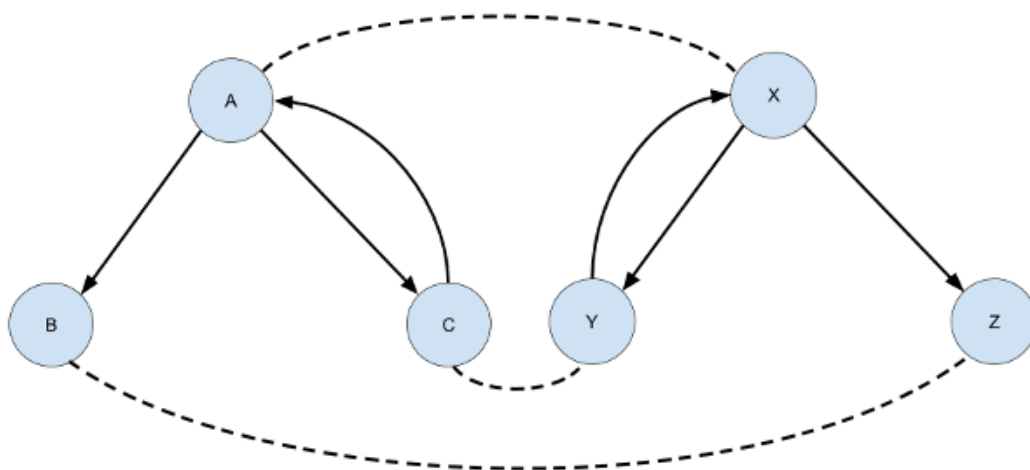


Figure 6.2: Variation in terms of student and an expert in causal map.

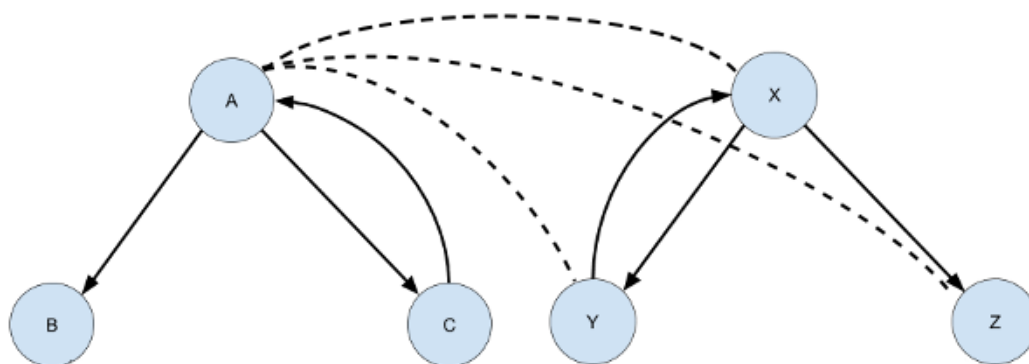


Figure 6.3: The possible number of comparison between a term of student's and expert map.

One of the key contributions of this software is to provide a visual way to align student's terms with expert terms by providing them with visual feedback (chapter 3) using an aggregate based recommendation system.

Once a common set of terms are established after alignment between two networks, multiple mathematical techniques are used to provide numerical measurements of how closely the two networks match. The second major contribution includes comparing a student causal map with an

expert's map using multiple graph algorithms like graph edit distance, graph kernel, and graph embedding.

Although convenient, a simple measurement does not provide much information about where a student could improve. To provide this type of information, we must examine the networks in more depth and identify features that need to be changed in order to improve the student's network. The third important contribution of this thesis includes providing hints or feedback regarding changes that should be made in student maps in order to bring them closer to causal maps provided by experts by finding loops and disjoint paths in the network (chapter 5).

## **6.2 Limitations**

One limitation of the current study is the requirement for individuals to accurately process visual cues. We assumed that the causal map provided by an expert will show all the concepts and factors influencing another factor. This might not hold true all the time and an expert might not predict all the factors that will influence a system. This limitation can be overcome if multiple reviews of the causal map are done.

Another limitation that we faced was a limited number of users and thus we had a small thesaurus of similar terms. This limits us in implementing a collaborative based recommendation system as collaborative filtering requires very large data sets, although this limitation will be fixed in a future enhancement when we will have enough term alignments done by users.

## **6.3 Future work**

### **6.3.1 Collaborative filtering based recommendation system**

Alignment of terms can be a time-consuming process even with a recommendation system because of the fact that the number of users is limited in the initial phase of software development. Once we acquire sufficient data and build a thesaurus the recommendation system will be improved by using the collaborative filtering technique. Collaborative filtering has a challenge with multiple synonyms that a student might use to refer similar terms, for example “children movie” and “children film” mean the same thing. This decreases the performance of a collaborative filtering based recommendation system. In this case, topic modeling can be used for grouping different words that belong to the same topic. Some of the topic modeling techniques that can be used include the Latent Dirichlet Allocation technique.

### **6.3.2 Clustering students**

In order to improve the student’s learning and understanding of a system, adaptive and innovative based learning should be attempted that will thus improve the quality of education. However, this requires feedback that needs to be provided to the instructor of how students are performing in class. This can be achieved by clustering students in groups. We proposed a method for clustering students based on multiple graph matrices such as graph edit distance, graph kernel, and graph



embedding. These graph matrices can be used to contrast distance between causal maps submitted by students to form clusters. The prototype of this is implemented as shown in Figure 6.4.

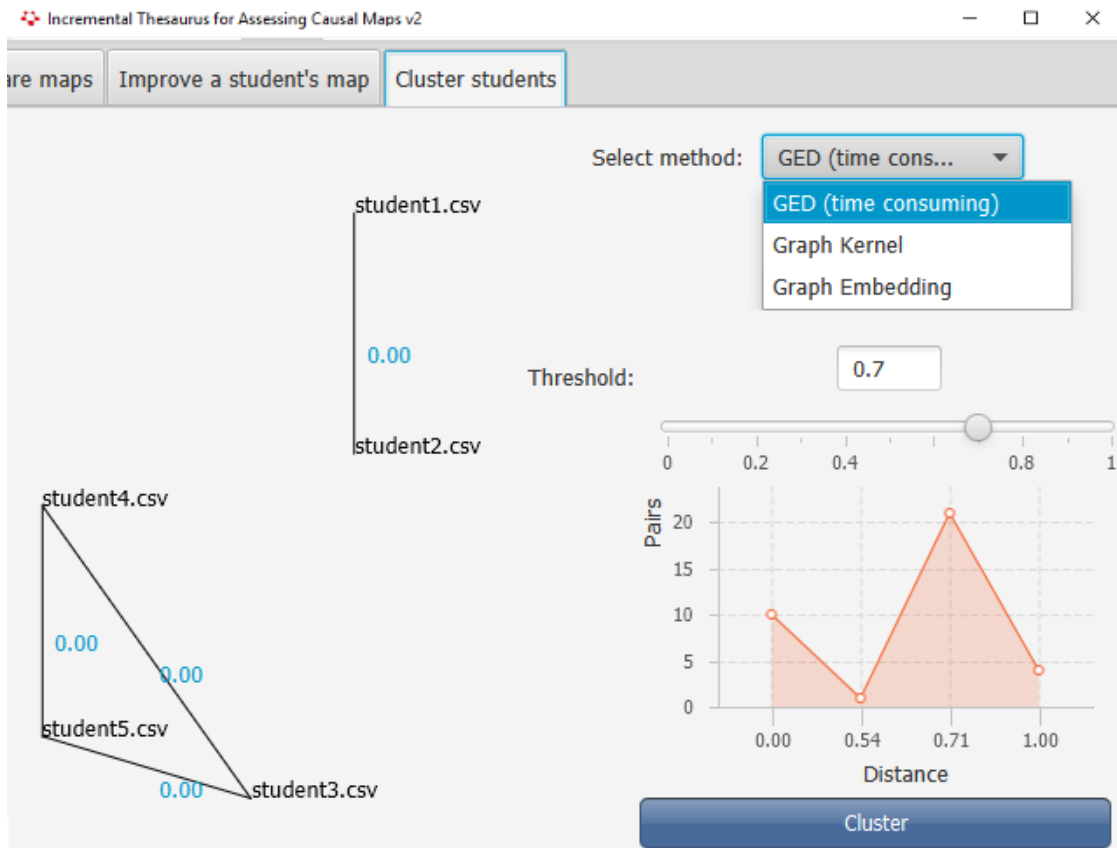


Figure 6.4: Clustering students based on different graph matrices.

## BIBLIOGRAPHY

- [1] C. C. Aggarwal. *Content-Based Recommender Systems*, pages 139–166. Springer, Cham, 2016.
- [2] A. Agrawal, J. Venkatraman, S. Leonard, and A. Paepcke. *YouEDU: Addressing Confusion in MOOC Discussion Forums by Recommending Instructional Video Clips*. Stanford InfoLab, February 2015.
- [3] E. K. Anders. Recent advances in expertise research: A commentary on the contributions to the special issue. *Applied Cognitive Psychology*, 19(2):233–241, 2005.
- [4] V. Arvind et al. Isomorphism testing: Perspective and open problems. *Bulletin of the European Association for Theoretical Computer Science*, 86:66–84, 2005.
- [5] D. G. Ausubel. Cognitive structure and the facilitation of meaningful verbal learning. *Journal of Teacher Education*, 14(2):217–222, 1963.
- [6] R. Axelrod et al. *Structure of decision: The Cognitive Maps of Political Elites*. Princeton University Press, Princeton, NJ, 1976.
- [7] H. S. Barrows. Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning*, 68:3–12, 1996.
- [8] M. Baur and M. Benkert. Network comparison. In U. Brandes and T. Erlebach, editors, *Network Analysis: Methodological Foundations*, pages 318–340. Springer, Berlin, 2005.

- [9] V. Carletti, P. Foggia, A. Saggese, and M. Vento. Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with VF3. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):804–818, April 2018.
- [10] K.-E. Chang, S. Yao-Ting, C. Rey-Bin, and L. Shui-Cheng. A new assessment for computer-based concept mapping. *Journal of Educational Technology & Society*, 8(3), 2005.
- [11] R. B. Clariana. Deriving individual and group knowledge structure from network diagrams and from essays. In *Computer-Based Diagnostics and Systematic Analysis of Knowledge*, pages 117–130. Springer, Boston, 2010.
- [12] R. B. Clariana, T. Engelmann, and W. Yu. Using centrality of concept maps as a measure of problem space states in computer-supported collaborative problem solving. *Educational Technology Research and Development*, 61(3):423–442, June 2013.
- [13] T. H. Cormen et al. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 3rd edition, 2009.
- [14] M. Dascalu, D. S. McNamara, S. Trausan-Matu, and L. K. Allen. Cohesion network analysis of CSCL participation. *Behavior Research Methods*, 50(2):604–619, April 2018.
- [15] M. Dehmer, L. A. J. Mueller, and F. Emmert-Streib. Quantitative network measures as biomarkers for classifying prostate cancer disease states: A systems approach to diagnostic biomarkers. *PLOS ONE*, 8(11):1–8, November 2013.
- [16] P. J. Dickinson, H. Bunke, A. Dadej, and M. Kraetzl. On graphs with unique node labels. In E. Hancock and M. Vento, editors, *Proceedings of the 4th IAPR International Workshop on Graph Based Representations in Pattern Recognition (GbRPR)*, pages 13–23, Springer, Berlin, 2003.

- [17] N. Dowell, O. Skrypnyk, S. Joksimovic, A. Graesser, S. Dawson, D. Gasevic, T. Hennis, P. de Vries, and V. Kovanovic. Modeling learners' social centrality and performance through language and discourse. *8th International Conference on Educational Data Mining (EDM)*, 2015.
- [18] L. Drasic and P. Giabbanelli. Exploring the interactions between physical well-being and obesity. *Canadian Journal of Diabetes*, 39:S12–S13, 2015.
- [19] R. J. Dufresne, W. J. Gerace, P. T. Hardiman, and J. P. Mestre. Constraining novices to perform expertlike problem analyses: Effects on schema acquisition. *Journal of the Learning Sciences*, 2(3):307–331, 1992.
- [20] A. Ertmer et al. How instructional design experts use knowledge and experience to solve ill-structured problems. *Performance Improvement Quarterly*, 21(1):17–42, 2008.
- [21] D. Eseryel, D. Ifenthaler, and X. Ge. Validation study of a method for assessing complex ill-structured problem solving by using causal representations. *Educational Technology Research and Development*, 61:443–463, June 2013.
- [22] A. Ezen-Can, K. E. Boyer, S. Kellogg, and S. Booth. Unsupervised modeling for understanding MOOC discussion forums: A learning analytics approach. In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge (LAK '15)*, pages 146–150, ACM, New York, 2015.
- [23] R. Ferguson and D. Clow. Examining engagement: Analysing learner subpopulations in massive open online courses (MOOCs). In *Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, LAK '15*, pages 51–58, ACM, New York, 2015.

- [24] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(1):1450001, 2014.
- [25] X. Ge and S. M. Land. Scaffolding students' problem-solving processes in an ill-structured task using question prompts and peer interactions. *Educational Technology Research and Development*, 51(1):21–38, March 2003.
- [26] F. Gedikli, D. Jannach, and M. Ge. How should I explain? A comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4):367–382, 2014.
- [27] P. J. Giabbanelli. Modelling the spatial and social dynamics of insurgency. *Security Informatics*, 3(1):2, 2014.
- [28] P. J. Giabbanelli, J. Adams, and V. S. Pillutla. Feasibility and framing of interventions based on public support: Leveraging text analytics for policymakers. In G. Meiselwitz, editor, *Social Computing and Social Media: 8th International Conference, SCSM 2016 Proceedings*, pages 188–200, Springer, Cham, 2016.
- [29] P. J. Giabbanelli, G. J. Babu, and M. Baniukiewicz. A novel visualization environment to support modelers in analyzing data generated by cellular automata. In V. G. Duffy, editor, *Digital Human Modeling: Applications in Health, Safety, Ergonomics and Risk Management: 7th International Conference, DHM 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings*, pages 529–540, Springer, Cham, 2016.
- [30] P. J. Giabbanelli and R. Crutzen. Creating groups with similar expected behavioural response in randomized controlled trials: A fuzzy cognitive map approach. *BMC Medical Research Methodology*, 14(1):130, December 2014.

- [31] P. J. Giabbanelli and P. J. Jackson. Using visual analytics to support the integration of expert knowledge in the design of medical models and simulations. *Procedia Computer Science*, 51:755–764, 2015. International Conference On Computational Science, ICCS 2015.
- [32] P. J. Giabbanelli and A. A. Tawfik. Overcoming the PBL assessment challenge: Design and development of the incremental thesaurus for assessing causal maps (ITACM). *Technology, Knowledge and Learning*, pages 1–8, September 2017.
- [33] N. Gillani and R. Eynon. Communication patterns in massively open online courses. *The Internet and Higher Education*, 23:18–26, 2014.
- [34] S. Goggins, K. Galyen, E. Petakovic, and J. Laffey. Connecting performance to social structure and pedagogy as a pathway to scaling learning analytics in MOOCs: An exploratory study. *Journal of Computer Assisted learning*, 32(3):244–266, 2016.
- [35] S. Gray, J. Hilsberg, A. McFall, and R. Arlinghaus. The structure and function of angler mental models about fish population ecology: The influence of specialization and target species. *Journal of Outdoor Recreation and Tourism*, 12:1–13, 2015.
- [36] S. A. Gray, E. Zanre, and S. R. J. Gray. Fuzzy cognitive maps as representations of mental models and group beliefs. In E. I. Papageorgiou, editor, *Fuzzy Cognitive Maps for Applied Sciences and Engineering: From Fundamentals to Extensions and Learning Algorithms*, pages 29–48. Springer, Berlin, 2014.
- [37] T. A. Grotzer, A. M. Kamarainen, M. S. Tutwiler, S. Metcalf, and C. Dede. Learning to reason about ecosystems dynamics over time: The challenges of an event-based causal focus. *BioScience*, 63(4):288–296, 2013.

- [38] V. K. Gupta, P. J. Giabbanelli, and A. A. Tawfik. An online environment to compare students' and expert solutions to ill-structured problems. In *Proceedings of the Human-Computer Interaction Conference (HCI 2018)*, 2018.
- [39] J. He, J. Bailey, B. I. P. Rubinstein, and R. Zhang. Identifying at-risk students in massive open online courses. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 1749–1755. AAAI Press, 2015.
- [40] T. Hecking, S. Ziebarth, and H. U. Hoppe. Analysis of dynamic resource access patterns in online courses. *Journal of Learning Analytics*, 1(2):34–60, 2014.
- [41] S. Henly-Shepard, S. A. Gray, and L. J. Cox. The use of participatory modeling to promote social learning and facilitate community disaster planning. *Environmental Science and Policy*, 45(Supplement C):109–122, 2015.
- [42] J. Herrington, T. C. Reeves, and R. Oliver. Authentic learning environments. In J. Spector, M. Merrill, J. Elen, and M. Bishop, editors, *Handbook of Research on Educational Communications and Technology*, pages 401–412. Springer, New York, 2014.
- [43] K. Hew and W. Cheung. Students' and instructors' use of massive open online courses (MOOCs): Motivations and challenges. *Educational Research Review*, 12:45–58, 2014.
- [44] G. R. Hjaltason and H. Samet. Properties of embedding methods for similarity searching in metric spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):530–549, May 2003.
- [45] C. Hmelo-Silver, S. Marathe, and L. Liu. Fish swim, rocks sit, and lungs breathe: Expert-novice understanding of complex systems. *Learning Sciences*, 16:307–331, June 2007.
- [46] C. E. Hmelo-Silver. Creating a learning space in problem-based learning. *Interdisciplinary Journal of Problem-Based Learning*, 7(1), 2013.

- [47] C. E. Hmelo-Silver and H. S. Barrows. Goals and strategies of a problem-based learning facilitator. *Interdisciplinary Journal of Problem-Based Learning*, 1(1), 2006.
- [48] C. E. Hmelo-Silver and M. G. Pfeffer. Comparing expert and novice understanding of a complex system from the perspective of structures, behaviors, and functions. *Cognitive Science*, 28(1):127–138, 2004.
- [49] V. Ho, R. K. Kumar, and G. Velan. Online testable concept maps: Benefits for learning about the pathogenesis of disease. *Medical Education*, 48(7):687–697, 2014.
- [50] Y.-C. Hong and I. Choi. Three dimensions of reflective thinking in solving design problems: A conceptual model. *Educational Technology Research and Development*, 59(5):687–710, 2011.
- [51] W. Hung. Theory to reality: A few issues in implementing problem-based learning. *Educational Technology Research and Development*, 59(4):529–552, August 2011.
- [52] D. Ifenthaler. Bridging the gap between expert-novice differences. *Journal of Research on Technology in Education*, 43(2):103–117, 2010.
- [53] D. Ifenthaler. Relational, structural, and semantic analysis of graphical representations and concept maps. *Educational Technology Research and Development*, 58(1):81–97, February 2010.
- [54] D. Ifenthaler. Determining the effectiveness of prompts for self-regulated learning in problem-solving scenarios. *Educational Technology and Society*, 15:38–52, January 2012.
- [55] D. Ifenthaler, I. Masduki, and N. 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, January 2011.



- [56] D. Ifenthaler and P. Pirnay-Dummer. Model-based tools for knowledge assessment. In J. Spector et al., editors, *Handbook of Research on Educational Communications and Technology*, pages 289–301. Springer, New York, 2014.
- [57] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization '90*, pages 361–378, October 1990.
- [58] M. J. Jacobson. Problem solving, cognition, and complex systems: Differences between experts and novices. *Complexity*, 6(3):41–49, 2001.
- [59] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems: An Introduction*. Cambridge University Press, New York, 2010.
- [60] M. Jenders, R. Krestel, and F. Naumann. Which answer is best?: Predicting accepted answers in MOOC forums. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 679–684, 2016.
- [61] A. Jeong. Sequentially analyzing and modeling causal mapping processes that support causal understanding and systems thinking. In D. Ifenthaler and R. Hanewald, editors, *Digital Knowledge Maps in Education: Technology-Enhanced Support for Teachers and Learners*, pages 239–251. Springer, New York, 2014.
- [62] S. Jiang, S. Fitzhugh, and M. Warschauer. Social positioning and performance in MOOCs. *Workshop on Graph-Based Educational Data Mining*, page 14, 2014.
- [63] D. B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.
- [64] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*, 24(1):1–13, January 1977.

- [65] D. W. Johnson and R. T. Johnson. Cooperation and the use of technology. In D. Jonassen, editor, *Handbook of research on educational communications and technology*, pages 659–670. Routledge, New York, 2008.
- [66] D. H. Jonassen. Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research and Development*, 39(3):5–14, September 1991.
- [67] D. H. Jonassen. Toward a design theory of problem solving. *Educational Technology Research and Development*, 48(4):63–85, December 2000.
- [68] D. H. Jonassen. *Learning to Solve Problems: A Handbook for Designing Problem-Solving Learning Environments*. Routledge, 2010.
- [69] D. H. Jonassen. Design problems for secondary students. *National Center for Engineering and Technology Education*, 2011.
- [70] V. Jordão, D. Gonçalves, and S. Gama. Eduvis: Visualizing educational information. In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14, pages 1011–1014, New York, 2014. ACM.
- [71] K. Jordan. Initial trends in enrolment and completion of massive open online courses. *International Review of Research in Open and Distributed Learning*, 15(1), 2014.
- [72] H. Ju and I. Choi. The role of argumentation in hypothetico-deductive reasoning during problem-based learning in medical education: A conceptual framework. *Interdisciplinary Journal of Problem-Based Learning*, 12(1), 2017.
- [73] K. Juszczyszyn, P. Kazienko, and K. Musiał. Local topology of social network based on motif analysis. In I. Lovrek, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 97–105. Springer, Berlin, 2008.

- [74] J. Kehrer et al. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, March 2013.
- [75] K. Kim and R. B. Clariana. Knowledge structure measures of reader’s situation models across languages: Translation engenders richer structure. *Technology, Knowledge and Learning*, 20(2):249–268, 2015.
- [76] N. J. Kim, B. R. Belland, and A. E. Walker. Effectiveness of computer-based scaffolding in the context of problem-based learning for STEM education: Bayesian meta-analysis. *Educational Psychology Review*, 30(2):397–429, July 2017.
- [77] R. F. Kizilcec, C. Piech, and E. Schneider. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. ACM, 2013.
- [78] D. Koller, A. Ng, C. Do, and Z. Chen. Retention and intention in massive open online courses: In depth. *Educause Review*, 3(48):62–63, 2013.
- [79] K. Kotovsky, J. Hayes, and H. Simon. Why are some problems hard? Evidence from tower of hanoi. *Cognitive Psychology*, 17(2):248–294, 1985.
- [80] H. Krabbe. Digital concept mapping for formative assessment. In D. Ifenthaler and R. Hanewald, editors, *Digital Knowledge Maps in Education: Technology-Enhanced Support for Teachers and Learners*, pages 275–297. Springer, New York, 2014.
- [81] E. A. Lavin, P. J. Giabbanelli, A. T. Stefanik, S. A. Gray, and R. Arlinghaus. Should we simulate mental models to assess whether they agree? *Proceedings of the Annual Simulation Symposium (ANSS) at the Spring Simulation Multi-conference*, 2018.
- [82] A. W. Lazonder and R. Harmsen. Meta-analysis of inquiry-based learning: Effects of guidance. *Review of Educational Research*, 86(3):681–718, 2016.

- [83] W. Liu, Ł. Kidziński, and P. Dillenbourg. Semiautomatic annotation of MOOC forum posts. In Y. Li et al., editors, *State-of-the-Art and Future Directions of Smart Learning*, pages 399–408. Springer, Singapore, 2016.
- [84] S.-B. Maggi. Understanding the impact of assessment on students in problem-based learning. *Innovations in Education and Teaching International*, 41(2):221–233, 2004.
- [85] P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1):3–35, April 2009.
- [86] L. Malhi, Ö. Karanfil, T. Merth, M. Acheson, A. Palmer, and D. T. Finegood. Places to intervene to make complex food systems more healthy, green, fair, and affordable. *Journal of Hunger & Environmental Nutrition*, 4(3-4):466–476, 2009.
- [87] J. R. McClure, B. Sonak, and H. K. Suen. Concept map assessment of classroom learning: Reliability, validity, and logistical practicality. *Journal of Research in Science Teaching*, 36(4):475–492, 1999.
- [88] L. A. J. Mueller, M. Dehmer, and F. Emmert-Streib. Comparing biological networks: A survey on graph classifying techniques. In *Systems Biology: Integrative Biology and Simulation Tools*, pages 43–63. Springer, Dordrecht, 2013.
- [89] J. Novak and D. Gowin. Concept mapping for meaningful learning. In *Learning How to Learn*, pages 15–54. Cambridge University Press, New York, 1984.
- [90] A. Olney, A. C. Graesser, and N. K. Person. Question generation from concept maps. *Dialogue & Discourse*, 3:75–99, 2012.
- [91] G. J. Passmore. Extending the power of the concept map. *Alberta Journal of Educational Research*, 50(4), 2004.

- [92] D. Perera, J. Kay, I. Koprinska, K. Yacef, and O. R. Zaiiane. Clustering and sequential pattern mining of online collaborative learning data. *IEEE Transactions on Knowledge and Data Engineering*, 21(6):759–772, June 2009.
- [93] S. F. Pratt, P. J. Giabbanelli, and J. S. Mercier. Detecting unfolding crises with visual analytics and conceptual maps emerging phenomena and big data. In *2013 IEEE International Conference on Intelligence and Security Informatics*, pages 200–205, 2013.
- [94] J. Qiu, J. Tang, T. X. Liu, J. Gong, C. Zhang, Q. Zhang, and Y. Xue. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, (WSDM '16), pages 93–102. ACM, 2016.
- [95] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender Systems Handbook*, pages 1–35. Springer, Boston, 2011.
- [96] K. Riesen. Graph edit distance. In *Structural Pattern Recognition with Graph Edit Distance: Approximation Algorithms and Applications*, pages 29–44. Springer, Cham, 2015.
- [97] K. Riesen. Graph edit distance – Novel approximation algorithms. In H. Bunke, editor, *Handbook of Pattern Recognition and Computer Vision*, chapter 2, pages 275–291. World Scientific, Singapore, 5th edition, 2015.
- [98] K. Riesen, S. Emmenegger, and H. Bunke. A novel software toolkit for graph edit distance computation. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 142–151. Springer, Berlin, 2013.
- [99] K. Riesen, A. Fischer, and H. Bunke. Combining bipartite graph matching and beam search for graph edit distance approximation. In N. Gayar, F. Schwenker, and C. Suen, editors, *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, volume 8774, pages 117–128. Springer, Cham, 2014.

- [100] K. Riesen, M. Neuhaus, and H. Bunke. Bipartite graph matching for computing the edit distance of graphs. In F. Escolano and M. Vento, editors, *Graph-Based Representations in Pattern Recognition: 6th IAPR-TC-15 International Workshop, (GbRPR 2007) Proceedings*, pages 1–12. Springer, Berlin, 2007.
- [101] J. S. Risch, R. A. May, S. T. Dowson, and J. J. Thomas. A virtual environment for multimedia intelligence data analysis. *IEEE Computer Graphics and Applications*, 16(6):33–41, 1996.
- [102] C. Romero and S. Ventura. Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010.
- [103] L. A. Rossi and O. Gnawali. Language independent analysis and classification of discussion threads in Coursera MOOC forums. In *15th IEEE International Conference on Information Reuse and Integration*, pages 654–661. IEEE, 2014.
- [104] M. Ruiz-Primo et al. On the use of concept maps as an assessment tool in science: What we have learned so far. *REDIE Revista Electrónica de Investigación Educativa*, 2(1), 2000.
- [105] J. Sanz, J. Navarro, A. Arbués, C. Martín, P. C. Marijuán, and Y. Moreno. The transcriptional regulatory network of Mycobacterium Tuberculosis. *PLOS ONE*, 6(7):e22178, 2011.
- [106] G. S. Sarwar and D. L. Trumpower. Effects of conceptual, procedural, and declarative reflection on students’ structural knowledge in physics. *Educational Technology Research and Development*, 63(2):185–201, April 2015.
- [107] H. G. Schmidt, J. I. Rotgans, and E. H. Yew. The process of problem-based learning: What works and why. *Medical Education*, 45(8):792–806, 2011.

- [108] C. Schumacher and D. Ifenthaler. Features students really expect from learning analytics. *Computers in Human Behavior*, 78:397–407, 2018.
- [109] N. Sclater, A. Peasgood, and J. Mullan. Learning analytics in higher education: A review of UK and international practice, 2016.
- [110] S. Sen, T. J.-J. Li, W. Team, and B. Hecht. Wikibrain: Democratizing computation on Wikipedia. In *Proceedings of the International Symposium on Open Collaboration*, volume 27 of *OpenSym '14*, pages 1–10. ACM, 2014.
- [111] E. Senocak. Development of an instrument for assessing undergraduate science students' perceptions: The problem-based learning environment inventory. *Journal of Science Education and Technology*, 18(6):560–569, 2009.
- [112] A. Sher. Assessing the relationship of student-instructor and student-student interaction to student learning and satisfaction in web-based online learning environment. *Journal of Interactive Online Learning*, 8(2), 2009.
- [113] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt. Efficient graphlet kernels for large graph comparison. In *Proceedings of Machine Learning Research*, volume 5, pages 488–495, 2009.
- [114] L. Skyttner. The future of systems theory. In *General Systems Theory*, pages 485–504. World Scientific, 2nd edition, 2011.
- [115] A. Solé, F. Serratos, and A. Sanfeliu. On the graph edit distance cost: Properties and applications. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(05):1260004, 2012.
- [116] J. M. Spector. A methodology for assessing learning in complex and ill-structured task domains. *Innovations in Education and Teaching International*, 43(2):109–120, 2006.

- [117] J. M. Spector et al. Technology enhanced formative assessment for 21st century learning. *Journal of Educational Technology & Society*, 19(3):58–731, 2016.
- [118] G. Stahl, T. Koschmann, and D. Suthers. Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer, editor, *Cambridge Handbook of the Learning Sciences*, pages 409–426. Cambridge, United Kingdom, 2006.
- [119] J. Stasko, C. Görg, and Z. Liu. Jigsaw: Supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.
- [120] J. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, 1974.
- [121] J. W. Suurballe and R. E. Tarjan. A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2):325–336, 1984.
- [122] S. R. Tamim and M. M. Grant. Definitions and uses: Case study of teachers implementing project-based learning. *Interdisciplinary Journal of Problem-Based Learning*, 7(2):3, 2013.
- [123] A. A. Tawfik, P. J. Giabbanelli, M. Hogan, F. Msilu, A. Gill, and C. S. York. Effects of success v failure cases on learner-learner interaction. *Computers & Education*, 118:120–132, 2018.
- [124] A. A. Tawfik, T. D. Reeves, A. E. Stich, A. Gill, C. Hong, J. McDade, V. S. Pillutla, X. Zhou, and P. J. Giabbanelli. The nature and level of learner–learner interaction in a chemistry massive open online course (MOOC). *Journal of Computing in Higher Education*, 29(3):411–431, December 2017.
- [125] A. A. Tawfik, H. Rong, and I. Choi. Failing to learn: Towards a unified design approach for failure-based learning. *Educational Technology Research and Development*, 63(6):975–994, December 2015.



- [126] D. L. Trumpower, M. Filiz, and G. S. Sarwar. Assessment for learning using digital knowledge maps. In D. Ifenthaler and R. Hanewald, editors, *Digital Knowledge Maps in Education*, pages 221–237. Springer, New York, 2014.
- [127] D. L. Trumpower and T. E. Goldsmith. Structural enhancement of learning. *Contemporary Educational Psychology*, 29(4):426–446, 2004.
- [128] D. L. Trumpower and A. S. Vanapalli. Structural assessment of knowledge as, of, and for learning. In M. Spector, B. Lockee, and C. M., editors, *Learning, Design, and Technology: An International Compendium of Theory, Research, Practice, and Policy*, pages 1–22. Springer, Cham, 2016.
- [129] M. Vento. A long trip in the charming world of graphs for pattern recognition. *Pattern Recognition*, 48(2):291–301, 2015.
- [130] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, Aug. 2010.
- [131] A. Walker and H. Leary. A problem based learning meta analysis: Differences across problem types, implementation types, disciplines, and assessment levels. *Interdisciplinary Journal of Problem-based Learning*, 3(1):6, 2009.
- [132] K. Weinerth, V. Koenig, M. Brunner, and R. 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.
- [133] M. Wijnen, S. M. Loyens, G. Smeets, M. J. Kroeze, and H. T. Van der Molen. Students’ and teachers’ experiences with the implementation of problem-based learning at a university law school. *Interdisciplinary Journal of Problem-Based Learning*, 11(2):5, 2017.

- [134] P.-H. Wu, G.-J. Hwang, M. Milrad, H.-R. Ke, and Y.-M. 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.
- [135] E. H. Yew and K. Goh. Problem-based learning: An overview of its process and impact on learning. *Health Professions Education*, 2(2):75–79, 2016.