

Matching Knowledge Elements in Concept Maps using a Similarity Flooding Algorithm

Byron Marshall (*byron.marshall@bus.oregonstate.edu*), Hsinchun Chen, and Therani Madhusudan

Abstract

Concept mapping systems used in education and knowledge management emphasize flexibility of representation to enhance learning and facilitate knowledge capture. Collections of concept maps exhibit terminology variance, informality, and organizational variation. These factors make it difficult to match elements between maps in comparison, retrieval, and merging processes. In this work, we add an element anchoring mechanism to a similarity flooding (SF) algorithm to match nodes and substructures between pairs of simulated maps and student-drawn concept maps. Experimental results show significant improvement over simple string matching with combined recall accuracy of 91% for conceptual nodes and concept→link→concept propositions in student-drawn maps.

Key words: Semantic Matching; Concept Mapping; Semantic Networks; Conceptual Graphs; Computer Assisted Instruction

1 Introduction

People have encoded “models” of their knowledge in a vast number of documents. A variety of indexes connect concepts and documents to support information retrieval. Some indexes (e.g. a digital library’s topical index) are “model-rich” with carefully constructed conceptual taxonomies. Others, like the classic vector-space model, are “term-rich” tabulations of the documents’ terms. An information seeker also begins with a mental knowledge model [24] which needs to be matched to the index. Conceptual graphs can provide a bridge between term-rich/model-weak representations and the mental models used by information seekers, enabling more effective query expression, better indexing and retrieval, and better tools for integrating text and images.

Conceptual graphs (CG) associate concepts in node-link semantic representations consisting of “knowledge elements” such as labeled concept nodes, labeled relationship links, and small clusters of nodes and links. These structures have been used in education, business, and knowledge management [1-3, 5, 9, 11, 12, 14, 19, 24, 29]. For example, concept mapping is a well-regarded educational technique employing an informal CG structure. In the business domain, workflow systems encode processes as a CGs of entities and relationships. Text mining processes can extract CGs from collections of text, ontologies are expressed as CGs, and the semantic web promises to provide CG information to facilitate knowledge discovery. While these applications employ different levels of formality and

varying degrees of computerized support, they all record knowledge as a set of labeled nodes and links.

Comparing, retrieving, and merging are key operations for effectively managing a collection of conceptual graphs. Workflow systems, for example, need to compare business processes to identify overlapping steps, instructors review student-drawn concept maps by comparing them to expert knowledge, and related knowledge elements are retrieved from concept mapping systems to assist users. As ontologies change or as new ones are developed they need to be merged with existing structures and relations parsed from text need to be verified, organized, and combined to better represent extracted information. Comparison, retrieval, and merging all begin by matching individual elements between two graphs. Thus, algorithms able to effectively match items between graphs using nodes, links, and labels are useful in a variety of applications.

In this paper we develop an algorithm for matching knowledge elements in educational concept maps. This is an interesting testbed for at least two reasons. Firstly, Educational concept maps are minimally formal. Because they are designed to allow maximum expressiveness and application in a wide variety of topical domains, educational concept mapping systems rarely enforce naming rules for nodes or links. Therefore concept maps tend to be idiosyncratic: different people create different maps of the same topic [24]. This is interesting because people generally approach query tasks with similarly idiosyncratic mental models. Secondly, concept maps are relatively easy to obtain. Several computer-based concept mapping tools are widely available. Improved management tools for collections of concept maps would be potentially useful in a variety of educational applications. Section 2 reviews existing concept map applications, examines the associated computational challenges, and identifies similarity flooding (SF) as a potentially useful map processing algorithm. Our research questions are listed in Section 3 and our implementation of SF is described in Section 4. Sections 5 and 6 describe our simulated and student-drawn map experiments, while Sections 7 and 8 discuss our findings and identify promising future directions.

2 Literature Review and Background

2.1 Concept Map (CM) Applications

Educational concept mapping was developed to support a constructivist notion of meaningful learning [23]. Concept mapping clarifies a learner’s understanding by identifying key concepts and expressing key relations, and student maps can be used to focus instructional activities. Four key meaningful learning processes find significant expression in the concept mapping process and in the structural

organization of student concept maps: (1) new concept learning, (2) subsumption, (3) progressive differentiation, and (4) integrative reconciliation [24]. In concept mapping a learner identifies concepts, hierarchically organizes those concepts, differentiates between them, and expresses more complex cross-hierarchical relations. These cognitive processes are exhibited in multiple layers of hierarchical groupings and in links that connect separate parts of the hierarchical tree.

Empirical studies verify concept mapping's educational utility. Conceptual graphs are effective in cooperative interactions, as pre- and post-study aids, as a substitute for traditional text, and for updating and editing knowledge [6]. Knowledge-mapping training was shown to have positive text processing effects for university students even when maps are not explicitly used. Student concept map scores have also been compared to standardized test scores. Rye and Rubba surveyed map scoring methodologies and found a correlation between maps scores and performance on the standardized tests [27].

Concept maps are used in educational processes where students draw concept maps and instructors provide feedback to the students based on map content. Evaluating maps is a manual and tedious process [15]. Computerized concept mapping tools should (but do not) facilitate evaluation. Perhaps this is because we do not know which evaluation measures are both useful and feasible. A wide variety of map scoring techniques have been proposed and evaluated. Shavelson et al. identify not less than 128 possible ways of generating and scoring concept maps [28]. Proposed map evaluation measures include (1) quantitative measures of map characteristics such as the number of propositions, (2) structural measures such as the number of hierarchical levels found in the expressed relations, (3) correctness measures which reward the validity of a proposition, and (4) similarity to an expert map.

Chen, Lin and Chang evaluate student maps using fuzzy integration and fuzzy matching algorithms employing an extended concept map formalism in which concept maps are enhanced with importance rankings for each node and link [5]. After carefully constructing a master map from maps created by three experts, fuzzy matching was used to identify how closely each student's map resembled the master map. Student maps were constructed using a closed list of node names and relation types. Some correlation was found between a student's performance on a handwritten test and the similarity between the student's map and the master map. The correlation was more pronounced for high performing students and difficult subject matter. Accuracy was reduced when distinct but correct organizational structures were used in different maps; these variations were referred to as "cognitive shifts." It was suggested matching could be improved using (1) contextual information for nodes and links, and (2) a proposition-based matching mechanism.

Although concept maps are most frequently associated with education, they have also been used to assist with other types

of knowledge management. Researchers have investigated concept maps in hypermedia systems [12], concept maps as a part of a knowledge support system [11], and concept maps for collaboration through map sharing on the web [13]. The use of concept maps for information search and browsing has also been considered in more recent work [4]. Concept mapping has been used as part of a knowledge elicitation methodology intended to record, maintain, and exploit expert knowledge [2]. The CMapTools for concept mapping developed by IHMC (Institute for Human Machine Cognition) have been used in several topical domains including space exploration (<http://www.cmex.arc.nasa.gov>), medicine [9], Naval technician training [1], and meteorological investigation [14]. They have been combined with case-based reasoning techniques [3] and a map collection has been used to make suggestions as new maps are drawn [18]. Another study concludes that "concept mapping should be considered as the interface of choice to a knowledge repository to be used by master's students in Information Management" [29]. In short, the automatic processing of concept maps for KM continues to be a topic of interest in the KM research community.

Algorithms that find matching knowledge elements concept map pairs could increase the usefulness of concept map collections. Previous CM research has focused on overall map similarity without directly accounting for the differences of vocabulary and representation that commonly occur in human-created knowledge representations. This has been reasonably effective because educational research has frequently employed closed lists of concepts and KM studies generally involve domain experts who tend to share a common vocabulary. We note that the integration of multiple maps requires element matching rather than map similarity calculation. In education, matching elements is important to provide semi-automatic support for the cognitively-oriented map measures by matching the constructs in student and master maps to assess the correctness of drawn links, count levels of hierarchy, identify appropriate layers of progressive differentiation, and recognize cross-links connecting different parts of a generally hierarchical structure. An element-matching algorithm would support assessment by speeding up the process of assigning a score to a map and support teaching by speeding up instructional feedback.

2.2 Computational Challenges

Semantic integration has received a significant amount of attention in recent years, particularly from database researchers seeking to facilitate information sharing and integrate heterogeneous data sources [8]. Applications are appropriate for semantic integration when two key assumptions hold. (1) The application uses structured representations, and (2) it employs more than one representation [8]. Most available model matching algorithms rely on these assumptions. While computerized concept mapping tools employ structured representations at the implementation level, and while topics mapped by two different people will almost always be represented

differently, these assumptions are not really appropriate for collections of concept maps. The “structured” representations assumed in most semantic integration research are expected to contain domain appropriate semantics that embody some notion of the key entities existent in a domain of interest. The “schema” of a concept mapping system consists of “concepts” and “relations” and therefore does not have any direct correlation to the entities in any particular domain. Because the schema holding a set of concept maps lacks structural clues related to the important entities in a domain, terminology variation, informality, and organizational variations are especially problematic. Educational concept mapping systems generally have labeled nodes and labeled links with no designation of node types and no restrictions on link names. While previous research notes that concept map informality and organizational variations can confound computerized processes, terminology variations seem to have been largely ignored in existing concept map processing routines.

2.2.1 Terminology Variation

People often use different words to represent the same concepts or the same link types. Yet neither the CMapTools approach [3] nor Chen, Lin and Chang’s matching routines [5] directly address terminology variations. The CMapTools approach sidesteps terminology variation relying on nearby terms to establish overall similarity for a map pair and the maps used by [5] were constructed with a controlled set of nodes. Because educational concept mapping systems generally do not enforce controlled vocabularies, better approaches that deal with terminology variations are needed.

2.2.2 Informality

Kremer [16] notes that there is a dichotomy between a human user’s need to work with a flexible and forgiving (hence informal) system and the computer’s need for a (formal) system with strong semantics. Although concept maps are intuitive and more “computationally efficient” [17] than some other forms of presentation such as pure text or predicate logic [22], Kremer asserts that concept maps can be computationally enhanced by constraining the “types” of links and nodes that can be created. Leake et al. describe concept map informality by stating that “Concept maps appear similar to semantic nets but have no fixed semantics and vocabulary” [18]. Concept maps are described by Canas et al. as a “middle point” between structural representations of CBR cases and textual descriptions. “They include structural information and are intended to concisely represent key concept properties but may not use standardized semantics. This makes them more difficult to manipulate autonomously than standardized representations but also easier to acquire when domain experts are called upon to encode knowledge” [2]. Informality is a problematic but largely unavoidable characteristic of concept maps.

2.2.3 Organizational Variation

Constructivist learning theory asserts that there is no single correct representation of knowledge. This notion corresponds to the “cognitive shift” problem identified for

concept maps scoring [5]. Two people often represent the same concepts using different but equally correct structures. However, examples of organizational variation are not provided in previous literature. Section 3.13. identifies some common organizational variations we found in a large collection of human-drawn concept maps.

2.3 Matching Techniques

Given the computational challenges associated with concept maps, we will now consider how element matching might be performed. An obvious beginning is comparing node and link labels. Commonly available string matching routines evaluate the “cost” (often character-segment deletions, insertions, and transpositions) associated with converting one string into another. This approach has some obvious limitations. Link names are problematic because a link name is often repeated many times in a concept map and two maps will often label the same relation differently. Logical analysis could also be used for matching graphs. However, even though concept maps can support a variant of predicate logic, most (particularly those generated in an educational setting) lack the formality needed to support logic computations. Concept map matching challenges more closely resemble the problems faced by schema matching systems intended to establish mappings between elements in database schemas and conceptual models. They routinely address terminology variation and un-recognized formalism. Schema matching is a process which creates a mapping between elements of two schemas [26]. In the cited work a schema is defined as a set of elements connected by some structure. That definition clearly applies to concept maps. Schema matching, ontology matching, and representation matching are all used to describe systems or algorithms in this broad area of research. Rahm and Bernstein’s work includes a taxonomy covering many existing approaches. Choosing the best matching approach depends on the characteristics of the schemas, the matching environment, and the intended use of the resulting match.

Rahm and Bernstein’s taxonomy employs several classification criteria. This taxonomy differentiates various schema matching approaches on based information used and output characteristics. Table 1 summarizes the concept mapping implications of the various approaches to schema matching.

Another way of classifying schema matching algorithms is to contrast rule-based vs. learner-based functions. In rule-based systems hand-crafted rules appropriate to a particular domain or task guide the matching process. Two of the many examples of this kind of system are PROM [7] developed by Doan et al., and the PROMPT algorithm implemented as a module of Protégé-2000 [25]. Unfortunately, the informal nature of concept maps drawn as part of a learning process do not provide the kind of detailed validation information needed for rule-based matching. Learning-based methods depend on learnable patterns that persist across different map pairs. It is not at all clear that such patterns are present in student-drawn concept maps.

Table 1. Schema Matching for Concept Map Evaluation

Classification Criteria	Differentiating Characteristic	Concept Map Evaluation Implications
Instance vs. schema	Use of instance data	At the node and link level, each concept map represents its own “schema” with only one available instance therefore schema matching is generally more appropriate than instance matching
Element vs. structure	Matching granularity	Identification of concept→link→concept relations is important because meaningful learning depends on understanding the relationship between concepts and scoring techniques focus on relations
Language vs. constraint	Element similarity /differentiation algorithms	Little or no constraint information is available because educational concept mapping systems generally do not restrict entries in ways which are computationally useful
Matching cardinality	Match cardinality	Because organizational structures of student maps are highly varied, 1:1, 1:m, and m:1 are frequently appropriate match cardinalities for map elements although matches are much more commonly 1:1
Auxiliary information	Use of external resources to assist in matching	Educational concept mapping tools are likely to be used to map knowledge from a wide variety of domains so generic approaches are preferred

Based on our review of schema matching systems and an analysis of concept map characteristics, we suggest that an ideal matching system for concept maps would: (1) be schema rather than instance based, (2) allow a matching granularity of at least small map substructures such as concept→link→concept propositions, (3) be language-based because constraints are generally unavailable, (4) would support match cardinalities greater than 1:1 although 1:1 matches would be the norm, and (5) would rely on little or no auxiliary information to maximize generalizability. We propose a using an adapted similarity flooding algorithm because it meets these requirements.

2.4 Similarity Flooding

A seemingly appropriate matching algorithm called similarity flooding (SF) was proposed in [21]. It matches two directed graphs (schemas, catalogs, or other data structures) to produce a multi-mapping of corresponding nodes. Filters select the best mappings which are then manually reviewed. Algorithm effectiveness was measured in the original SF work by estimating the labor savings obtained using the algorithm for schema matching tasks. It is an inexact matching approach which relies on the intuition that elements of two graphs are similar when their adjacent elements are similar. SF is flexible and extensible because it requires only a general network representation to accomplish a match. Similarity and adjacency are generically defined making the algorithm usable for diverse matching tasks and tunable for multiple domains. The effectiveness of SF for structures other than database schemas has not been investigated.

SF promises to be useful for concept maps because it is able to simultaneously leverage both the link structure of a concept map and the semantic content of the node and link labels. Various granularities can be matched (individual nodes and node-link-node propositions), it does not rely on a

distinction between schema and instance data, and, although a variety of methods could be used to establish initial similarity, it is language-oriented. Many matching technique distinguish schema and instance information implicitly assuming that the schema structure contains a substantial amount of semantic information and that a preponderance of the instance-level data can be used as a differentiator. On face, these implicit assumptions do not seem appropriate for concept-map and concept-map-like applications. On the other hand, the basic intuition used by SF is that similar items will tend to be connected in different people’s maps. Intuitively, this assumption could be said to demand less precise structural similarity in the matched models and therefore be more appropriate for the kind of maps drawn in a classroom context.

3 Research Questions

Our review of the literature and our experiences with concept mapping in education suggest that the similarity flooding algorithm can be used to support concept map management processes by matching knowledge elements found in concept map pairs. We explore this potential by experimentally addressing three research questions:

1. Given query and target concept maps, can we correctly identify node and propositional knowledge elements from the query map in the target map?
2. How does the similarity flooding algorithm perform for this matching task?
3. How do commonly-observed concept map organizational variations affect the accuracy of the matching process?

We explore these questions in two experiments. The first experiment employed simulated concept maps to explore algorithm performance in a controlled setting. To implement our simulation, we evaluated a large collection of human-

drawn concept maps to establish a baseline for vocabulary overlap and reviewed topically-similar sets of maps to identify common organizational variations. Although these evaluations are somewhat peripheral to the main focus of this study, they may also be of interest to developers of concept mapping systems. The second experiment evaluated the algorithm using student-drawn concept maps. The balance of Section 3 describes our research testbed.

3.1 GetSmart System

Our understanding of the computational difficulties associated with student-drawn concept maps comes in part from our experiences building and using the GetSmart system [19]. GetSmart was developed at the University of Arizona as part of NSF's NSDL (National Science Digital Library) project with input from research partners at Virginia Tech. GetSmart supports educational processes by integrating course resources and advanced digital library technologies with concept mapping for personal knowledge representation. More than 100 students at the University of Arizona and Virginia Tech used GetSmart in the fall of 2002. Students created concept maps of course material individually and in groups. Concept maps are envisioned as a valuable part of an educational process in which students record their personal understanding and instructors provide feedback based on their expert knowledge. Use of the system in the fall of 2002 resulted in a concept map collection containing 30 or more maps for each of 11 subtopics related to data structures and algorithms, 30 or more maps for each of 10 chapters of an information retrieval textbook, and hundreds of other student-drawn concept maps. Table 2 provides usage statistics for GetSmart in the fall of 2002. The time spent evaluating and managing the maps in this large collection points to the real need for improved map processing algorithms.

Table 2. GetSmart Usage

Students at the University of Arizona and Virginia Tech created an extensive collection of concept maps.

114	Student Users
4,000 +	User Sessions
1,400 +	Maps Created; Homework & Presentations
50 +	Maps created by groups
40,000 +	concept→link→concept propositions in maps

3.2 Observed Vocabulary Overlap

To better understand the difficulties associated with concept matching, we did some initial analysis of our concept map collection to identify the degree of vocabulary overlap and to compile a list of observed structural variations. This information was used to guide development of the algorithm and simulations used in this work. Previously, [10] measured term overlap finding that people generally choose the same term for the same object about 20% of the time. Because our concept map collection was generated in a classroom setting by students who had all been exposed to the same instructional materials, we expected to see more overlap than was found in the cited study. To measure the overlap in our

collection, we chose 4 topical sets of maps and then randomly chose 10 maps from each topical set. We compared the different representations used by different people for the same concept. In some cases the user intended to use the same words for a concept but entered them in the concept map differently. For example the term "E Measure" was entered as (1) "E-Measure," (2) "emeasure," (3) "E Measure," (4) "E Measrue," and (5) "E evaluation measure." All of these representations were considered to represent the same concept. The first 4 were considered to be matching representations while the last was considered to be only somewhat similar. After making some minor spelling adjustments, we found that the most common word for a concept appeared in about 75% of the concept maps. About half of the remaining words were quite similar and the rest were not very similar at all. Using the measurement methodology described in Furnas et al., this equates to between 50 and 60% overlap. This seems to be a reasonable initial estimate of vocabulary overlap for concepts in student-drawn maps.

3.3 Observed Organizational Variations

In order to establish the usefulness of a concept map matching system, we needed to look at specific examples of the kinds of organizational variations (referred to as "cognitive shifts" in [5]) observed in a collections of topically-similar concept maps. Based on our review of the student-drawn maps in our collections, we identified four categories of variation: (1) missing elements, (2) added elements, (3) cross-links, and (4) other organizational variations. Without commenting on the correctness of the knowledge expressed in the map snippets, we observe that structural variations are associated with student understanding. Figures 1, 2, and 3 are all based on actual student-drawn concept maps covering tree data structures. Adding and missing elements are depicted in Figure 1. Leaving out important leaf nodes might reflect a student's failure to remember a key concept. Addition of internal nodes can express higher degrees of progressive differentiation. The first map allows for the addition of non-ordered tree traversal methods, the second depicts an additional concept. Figure 2 shows frequently seen organizational variations. In the first map, three types of binary trees are connected to the higher order concept *binary trees* in a simple hierarchical structure. In the second, a student uses linear arrangement implying subsumption relationships. The last map reverses the direction of the arrows and shows additional relationships. Figure 3 depicts cross-links. In most maps of the topic, students clustered general tree terminology as seen in the *children*, *parent*, and *sibling* nodes. These same maps also usually contain some representation of binary tree types. In Figure 3 two hierarchical sections are connected by cross-links. Novak and Gowin suggest that this indicates creativity and should be particularly rewarded in concept map scoring.

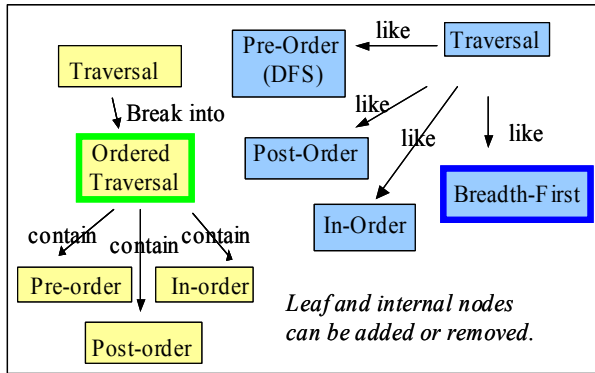


Fig. 1. Added or Missing Elements

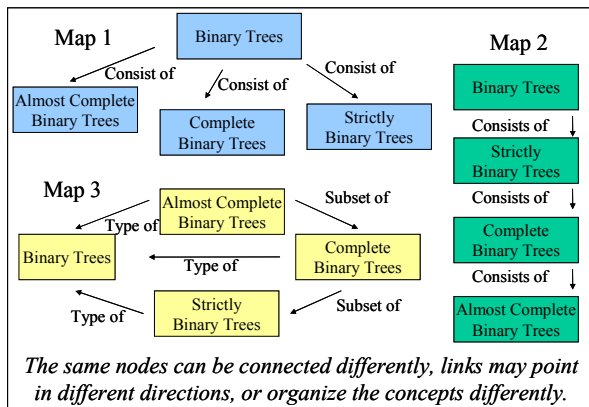


Fig. 2. Organizational Variations

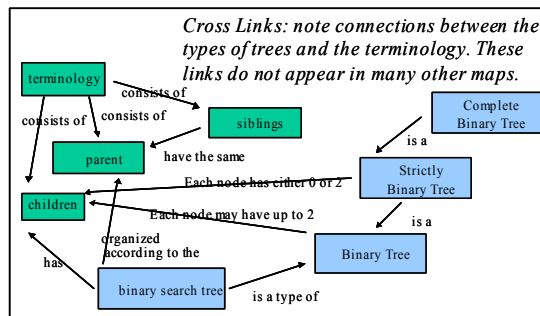


Fig. 3. Cross-Links

4 Implementation

Our similarity flooding algorithm closely parallels the initially proposed algorithm [20] but also features two knowledge-anchoring adaptations which address the algorithm's tendency to identify superstructure matches. The algorithm has 4 steps: (1) graph representation, (2) calculation of initial similarity, (3) a fixpoint computation, and (4) filtering. To illustrate the algorithm's function, the top of Figure 4 shows two small concept maps and the bottom shows how they might be represented during step 1 of the flooding algorithm. Links can also be represented as nodes when preparing the graph for the SF system. This

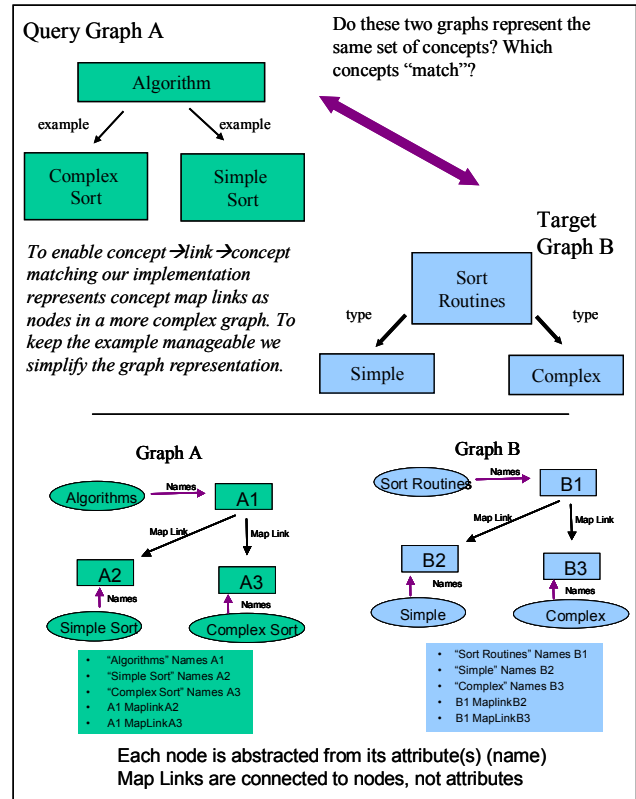


Fig. 4. Similarity Flooding Map Representations

extension would support matching at the concept→link→concept proposition level. Nodes and node names are abstracted as separate elements. Other available and appropriate attributes can also be attached to the nodes. This allows various attributes to independently contribute to the similarity calculation.

Next the algorithm obtains initial similarity values for Graph A/Graph B node pairs. Because our example maps each include 6 nodes (as shown in the bottom half of Figure 4) 36 initial similarity values can be provided. We provided initial similarity only for names (represented as ovals in the figure) and zero values are ignored. Although the initial values strongly affect matching accuracy, the other operations are independent of the initial similarity assignment process. Table 3 shows the values used in our example. No similarity is provided for the pair *Algorithms/Sort Routines*, and similarity is provided for the incorrect matches *Simple Sort/Sort Routines* and *Complex Sort/Sort Routines*. This emulates the results a string matching routine would provide. Once the graphs are represented and initial similarity values are established, an iterative fixpoint calculation that updates similarity based on adjacent node similarity creates a mapping between elements. A pairwise connectivity graph is generated, then the algorithm iterates using a fixpoint computation to pass similarity between node pairs until the network stabilizes. A formal description of the algorithm with its internal representation is available [20]. The algorithm operates on the assumption that whenever two elements, one from Graph 1 and one from Graph 2, are

found to be similar the similarity score of adjacent elements should be increased. Over a number of iterations, the initial similarity of any two nodes propagates through the graphs. In our example, the similarity initially identified for the node label pairs *Simple Sort/Simple* and *Complex Sort/Complex* propagate to the node pair “A1/B1” to establish the correct final match shown in Table 4. Figure 5 shows several of the paths along which the similarity travels from names, through node pairs to other node pairs. The he computation maps each Graph 1 element to every element in Graph 2.

Table 3. Example Initial Similarity Values

Node Pair	Initial Similarity Value Assigned
Simple Sort / Sort Routine	.5
Complex Sort / Sort Routine	.5
Simple Sort / Simple	.7
Complex Sort / Complex	.7

Table 4. Similarity Output – A Multimapping

Graph A Node	Graph B Node	Similarity Output
A1 (Algorithms)	B1 (Sort Routines)	1.00
A1	B2 (Simple)	0.10
A1	B3 (Complex)	0.10
A3 (Complex Sort)	B3 (Complex)	0.65
A3	B2 (Simple)	0.40
A3	B1 (Sort Routines)	0.07
A2 (Simple Sort)	B2 (Simple)	0.61
A2	B3 (Complex)	0.43
A2	B1 (Sort Routines)	0.15

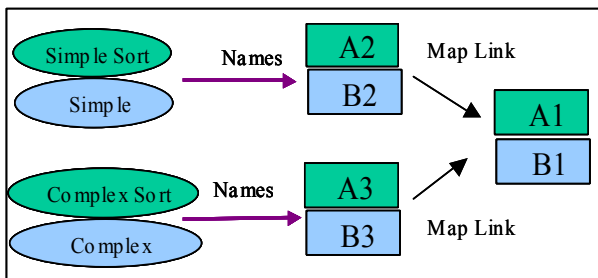


Fig. 5. Selected Propagation Graph Paths

Because this multimapping is too large for most applications, the fourth and final step chooses which matches to report. Three filters tested by Melnik et al. resulted in comparable accuracy: a *Threshold* filter chooses all matches above some threshold value, an *Exact* filter reports the highest match for each node in Graph 1, and a *Best* filter requires that each node in Graph 1 can be matched to only one node in Graph 2. The *best* filter uses a greedy algorithm where, for the next unmatched element, a best available candidate is chosen to maximize cumulative similarity. The highest accuracies were reported for the *threshold* and *exact* filters with only slightly lower results reported for the *best* filter. In our example, the highlighted

rows in Table 4 represent a correct one-to-one mapping and would be chosen by either the exact or best filters.

Melnik et al. evaluate four variations of the fixpoint calculation. *Basic*, *A*, *B*, and *C* are shown in Table 5. The function f increments the similarity of an element pair (σ^{i+1}) based on the similarity of its neighbors. The relative influence of the initial similarity value (σ^0) and the previous iteration's value (σ^i) changes in each variation. *C* is most strongly influenced by the initial similarity values. *Basic* was found to be the slowest to converge and the least accurate. *A*, *B*, and *C* had comparable convergence properties but *C* was slightly more accurate. In the graph representation phase each concept→link→concept proposition found in the map was presented to the algorithm three ways. First a node name and maplink relations are created (as shown in Figure 4). Then separate link elements are created and connected to the nodes to represent each proposition.

In initial tests we tended to match to “superstructures.” That is, viewing the graph as a somewhat hierarchical structure, we occasionally had incorrect matches to items at “higher” levels in the map structure. This is a documented tendency in the SF algorithm [21]. We addressed this problem by generating “hierarchical structure” elements when one node was connected to three or more nodes of the same color by links with the same name and direction, as shown in Figure 6. Existing concept maps, as drawn by students, require no special input or manual adjustments to identify these structures. We also introduced a “node anchoring” mechanism. Key terms and commonly used abbreviations are identified as anchor points. Whenever these terms are found in both query and target maps, they are “locked-in” as best matches. We increase the match value for these pairs in each iteration of the fixpoint computation.

Table 5. Fixpoint Formulas

Identifier	Fixpoint Formula
Basic	$\sigma^{i+1} = \text{normalize}(\sigma^i + f(\sigma^i))$
A	$\sigma^{i+1} = \text{normalize}(\sigma^0 + f(\sigma^i))$
B	$\sigma^{i+1} = \text{normalize}(f(\sigma^0 + \sigma^i))$
C	$\sigma^{i+1} = \text{normalize}(\sigma^0 + \sigma^i + f(\sigma^0 + \sigma^i))$

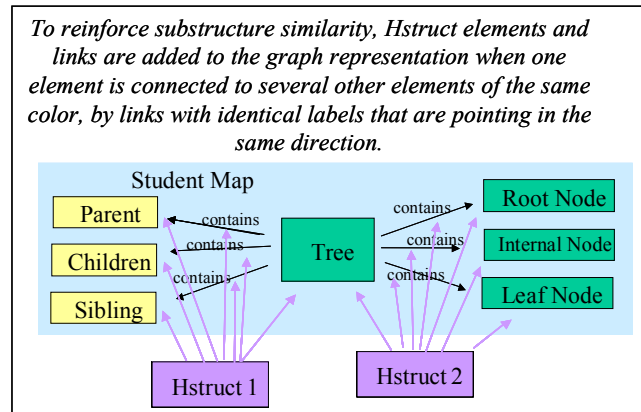


Fig. 6. Reinforcing Substructures

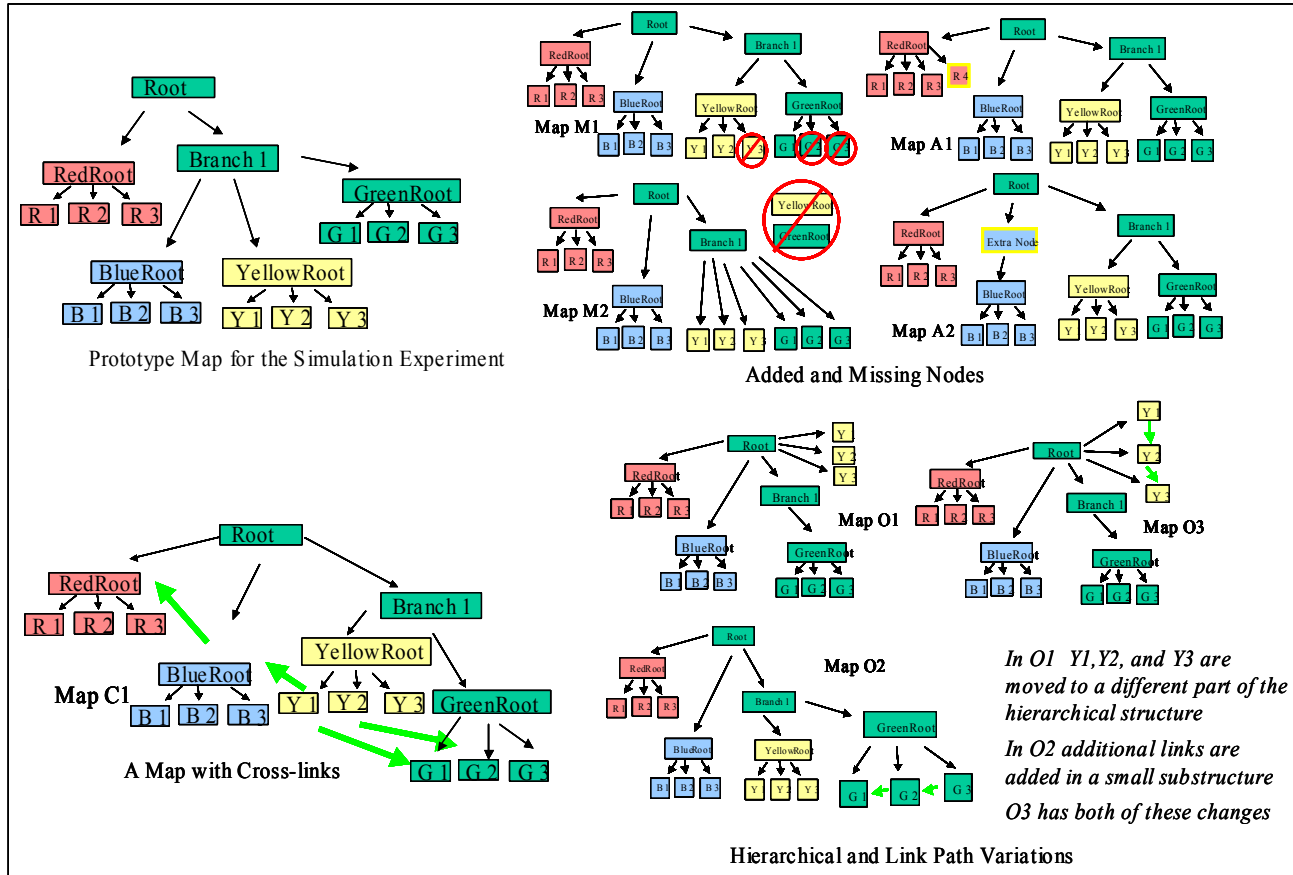


Fig. 7. Maps used in the Simulation

5 Simulation Experiment

Our simulated map experiment evaluates SF’s capabilities for matching elements in concept maps under different structural and terminological variations. First we verify functionality with concept maps because the algorithm was previously used on data schemas. We wanted to see if the algorithm increased concept map element matching accuracy above the initial similarity evaluation. Next we evaluated filtering methods and fixpoint formulas. Finally, using a set of systematically altered maps and 30 sets of randomized initial similarity values, we evaluated performance in the presence of organizational structure variations.

5.1 Simulation Experimental Design

The simulated map set is representative of a collection of human-drawn concept maps. Section 2.2 identifies three crucial concept map characteristics with implications for computational performance: informality, structural variation, and terminological variation. To represent a concept map collection we began with the map shown in Figure 7. The prototype is intended to represent an instructor-created expert or master map. Its nodes are named generically so it can represent a map on any topic. One of the key notions of the simulation is separation of the effects of map structure variation from the effects of initial node similarity values. By naming the nodes generically and assigning randomized

initial similarity values we allowed the comparison of these effects. The node labeled “Root” might represent the top-level concept for a map on any topic. The cluster of yellow nodes (YellowRoot, Y1, Y2, and Y3) stands for some grouping of important concepts related to the “Root” concept. The rest of the maps in the simulation are intended to mimic commonly observed, cognitively important organizational variations.

We compared the prototype map to 9 simulated maps. In maps A1, A2, M1, and M2 we added or removed nodes. Map C1 includes cross-links connecting different portions of the hierarchy. In O1 we attached the Y1, Y2, and Y3 nodes to the root node. In O2 we added additional links between nodes within a cluster and in O3 we made both of these changes together. Finally, we used an exact copy of the prototype map.

The second part of our simulated environment is a set of initial similarity values designed to simulate string match results from a set of concept maps drawn by different people on the same topic. Each comparison requires a matrix relating each node in the query map to each node in the target map. The matrix contains initial similarity values for nodes representing the same concept and values for nodes representing different concepts. For example, consider the node labeled *GreenRoot* in the simulated map. The simulated map M1 also has a node labeled *GreenRoot*. If the

prototype map were an instructor’s master map (query map Q) and map M1 was a student’s map (target map T), the student might have used the same term exactly, a very similar term, or a completely different term to represent *GreenRoot*. A correct result matches *GreenRoot* from the query map to the *GreenRoot* node in the target map.

To calculate values for same-concept node pairs (e.g. GreenRoot in map Q to GreenRoot in map T) we used the observed variance from Section 3.2. Table 6 describes the distribution of these values. An initial similarity value of 1 was assigned to 75% of the same concept nodes to represent those occasions when the same term is used to denote the same concept. A value of .85 was assigned to 15% to represent very similar representations. The rest were assigned .173 to simulate cases where different terms have been used. This distribution approximates the 50-60% vocabulary overlap found for same-concept nodes in a set of student-drawn maps. Table 7 describes the initial similarity value distribution used for nodes representing different concepts (i.e. GreenRoot in some map Q to R1 in some map T). We created this distribution using string match calculations performed on more than 25,000 node label pairs found in a set of 60 topically-similar concept maps. Figure 8 shows a portion of the initial similarity value matrix for one of the 30 computations for altered map A1.

Table 6. Simulated Initial Similarity for Same Concept Node Pairs

Simulated Map Nodes		Examples of What the Simulation Represents	Possible Initial Similarity Value
Node Pair	Node		
Prototype Map Node	Target Map Node		
R1	R1	Pre-Order, pre-order	1
R1	R1	Pre-Order, pre-order (DFS)	0.85
RedRoot	RedRoot	Traversal Method, Algorithm	0.175

Simulated matching nodes are assigned initial similarity based on analysis of the vocabulary overlap found in 40 maps covering 4 topics (equates to 50-60% overlap)

75% - Match Value 1.0	Nearly Exact Terminology
15% - Match Value .85	Very Similar Terminology
10% - Match Value .175	Significantly Different Terminology

Table 1. Simulated Initial Similarity for Different Concept Pairs

Simulated Map Nodes		Examples of What the Simulation Represents	Possible Initial Similarity Value
Node Pair	Node		
Prototype Map Node	Target Map Node		
R1	R2	Pre-Order, post-order	0.85
R1	Y3	Pre-Order, parent	0.51
Root	G3	Tree, Sibling	0

Simulated non-matching nodes are assigned initial similarity based on an analysis of string match values from 25,000+ node label pairs found in a set of 60 topically-similar concept maps

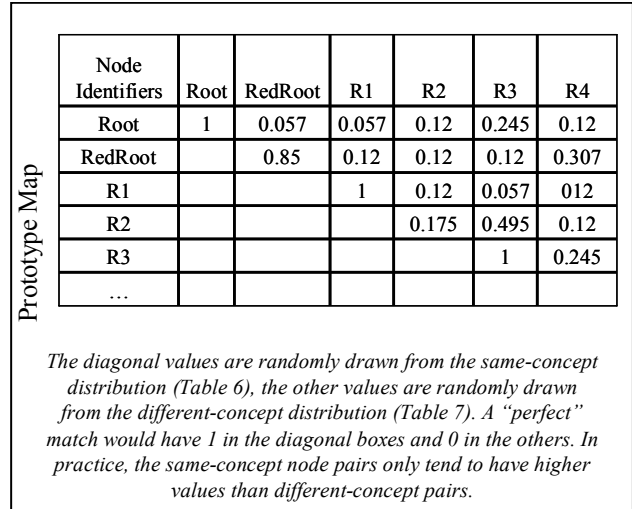


Fig. 8. Initial Similarity Matrix for Map A1

Each of the simulated maps was tested with 30 different sets of randomized initial similarity values. That is, each generated target map was compared to the prototype map 30 times. In the second step of the similarity algorithm, initial similarity for same-concept node pairs was assigned values from the first distribution, while values from the second distribution were used for different-concept pairs.

The similarity flooding (SF) algorithm performed as expected for the different fixpoint formula options. Formulas A, B, and C showed comparable recall accuracy, substantially out-performing the Basic formula as shown in Figure 9. Accuracy was measured by dividing the number of correctly matched nodes by the number of correct matches possible. Figure 10 shows that, as in the previous SF study, the algorithm converged more quickly (in fewer iterations) when employing Formula C.

The Best and Exact filters were compared because they both produce at most one match in the target graph for each node in the query graph while the Threshold filter allows multiple results for each query node. Employing the Best filter improved matching accuracy over Exact filtering. Best improved 132 of 240 mappings (55%) of those matches while reducing the accuracy of only 24 (10%). Best produced a net increase of 210 correctly matched nodes on 4,650 attempts (4.5%). Table 8 shows the improvement in node match recall for the best filter as compared to the exact filter. All the improvements were significant at the p=.05 level.

The similarity flooding algorithm improved on the node matching accuracy over a match based only on the initial similarity values as shown in Table 9. To create a comparable result, we used the Best filter for both the similarity flooding and string match results. Improvement ranged from 3 to 11 percent; the improvement in the average accuracy for the 30 trials of each of the 9 maps was found to be significant at the p=.05 level.

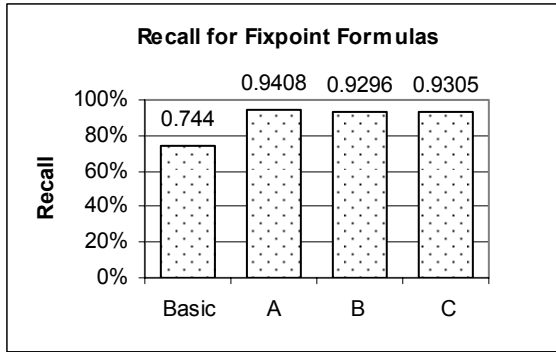


Fig. 9. Fixpoint Formula Accuracy

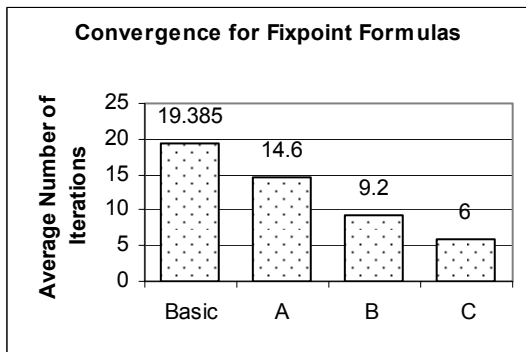


Fig. 10. Fixpoint Formula Convergence

Table 8. Filter Recall Results

	Node Recall		
	Best	Exact	Improvement
Nodes	.92	.88	4 %
Links	.88	.78	10 %
Elements	.93	.83	10%

Table 9. Correct Node Match Ratio, Similarity Flooding vs. Initial Similarity

Map Variation	SF Result	Initial Value	Improvement
Identical Graph	0.95	0.84	11%
A1 Added Leaf Node	0.94	0.85	9%
A2 Added Internal Node	0.91	0.84	7%
M1 Missing Leaf Nodes	0.94	0.86	8%
M2 Missing Internal Nodes	0.86	0.83	3%
C1 Cross-links	0.95	0.86	9%
O1 Moved Node Group	0.91	0.83	8%
O2 Added Links in a Substructure	0.96	0.89	7%
O3 Two Organizational Variations	0.93	0.84	9%

6 Student-Drawn Map Experiment

The student map experiment tested the algorithm in a more complex and realistic environment to see if a similarity flooding match was better than a match based on actual string match values. We also wanted to identify situations in which the algorithm returned inaccurate matches to identify strategies for improvement.

6.1 Student-Drawn Map Experimentation

Thirty topically similar, student-drawn target maps from the GetSmart collection were selected. They exhibited a variety of terminological and structural variations. We created a query map of the topic emphasizing hierarchical elements frequently found in the student maps. Hierarchical relations were emphasized because of their cognitive and educational importance. Two graduate students familiar with the topic compared the query map to each of the target maps. A list of correct matches was compiled; only matches agreed upon by both reviewers were used in calculating accuracy results. The query map was matched to each of the student maps using our similarity flooding implementation. The *Best* filter was used in both similarity flooding (SF) and string match (SM) processing. A commonly available string match algorithm provided similarity scores for each node in the query map to every node in each of the target maps. In addition, we compared the full text of each concept→link→concept proposition pair. For example the node label pair [*Traversal, Ordered Traversal*] was assigned an initial similarity of .529 by the string matching algorithm, and the proposition pair [*Ordered Traversal contains In-Order, Traversal include In-Order*] was assigned .559.

6.2 Student-Drawn Map Results

The similarity flooding (SF) based matching system outperformed the string matching (SM) algorithm for both concept nodes and concept→link→concept propositions. SF & SM were somewhat complementary. SM occasionally identified correct matches missed by the SF algorithm. Table 10 compares the SM and SF recall. SF+SM reflects occasions when either the SM or the SF result was correct. All of the SF results are significantly better than the SM results at the p=.05 level. Recall is the more important measure of accuracy because it would be relatively easy for an instructor to ignore incorrect matches when evaluating a map. However, recall can also be measured for various levels of precision. A minimum similarity threshold would cause the algorithm to ignore many poor matches. Recall at various precision levels is reported in Figure 11. By selecting an output similarity threshold of .8 we achieved both recall and precision above .90.

Table 10. Score of SF vs. SM for 30 Student-Drawn Concept Maps

	SF	SM	Improvement	SF+SM
Nodes	.94	.88	6%	.95
Propositions	.79	.50	29%	.84
Combined	.88	.72	16%	.91

7 Discussion

Although the matching results are encouraging, they can still be improved. We reviewed the remaining matching and identified four recurring problems. Incorrect matches can be traced to (1) student misconceptions (2) synonymy, (3) cardinality, and (4) granularity. Because informality and

flexibility enhance the educational value of concept mapping, our observations are intended to identify methods of increasing match accuracy without imposing restrictions on the student map building process.

Student misconceptions are exhibited in incorrect links and lack of organizational clarity. Some matching errors could be directly traced to factually incorrect links that introduced noise in the algorithm. Removing incorrect links (for example a link identifying a BTree as a type of Binary Tree) would have increased node matching accuracy in some cases. Map clarity is also important. Flat trees with few hierarchical levels reflect a lack of conceptual differentiation. Identifying such ambiguous or incorrect representations would be helpful educationally and increase matching accuracy.

Different people frequently use different abbreviations, synonyms, or word forms in node labels. Examples include CBT for Complete Binary Tree, Routines for Algorithms, and Trees vs. Tree. In many cases the system corrects these errors, but not always. For example, the term pairs child/descendant and parent/ancestor were presented in lecture as contextually equivalent terms for a concept. The string match algorithm matched child/ancestor over child/descendant, and parent/descendant over parent/ancestor. Because the concepts were placed in equivalent structural positions, the SF algorithm could not correct the match. Even when a correct match is found for such a node pair, ambiguous signals may be introduced into the similarity propagation graph causing other errors. A query map could be structured to include domain appropriate synonyms to help with these easily-anticipated problems and lexical resources could be used. We observed that locking in matches for key terms (i.e. "tree" always matches with "trees" and "binary tree" always matches with "binary tree") increased overall matching accuracy. Introducing just these two key terms for enhanced matching resulted in a substantial increase in overall matching performance in one set of maps. Key top-level terms could be listed in the query map or inferred from a map collection prior to processing the individual map pairs.

Matching cardinality also affects matching accuracy. We found that Best filtering (which enforces a 1:1 match cardinality) improved accuracy. But, our initial review of appropriate characteristics for a matching algorithm for concept maps (presented in Section 2.5) noted that 1:M and M:1 would be appropriate on some occasions. Out of 30 maps on tree structures from the GetSmart collection, 3 maps included Btrees and B+Trees in the same node B/B+Trees. The fixpoint calculation portion of the SF algorithm gave a high similarity score to both "Btree" \rightarrow "B/B+Tree" and to "B+Tree" \rightarrow B/B+Tree but the Best filter forced it to choose only one of the matches. In addition to the main error (missing the match between Btree and B/B+Tree), the resulting ambiguity occasionally caused other mismatches.

In this initial implementation we restricted granularity by allowing matches only between nodes and nodes or

propositions and propositions. In a number of cases, 3-element knowledge structures (node \rightarrow link \rightarrow node propositions) would have been better matched to 5 element structures (node \rightarrow link \rightarrow node \rightarrow link \rightarrow node). For example the query map includes the proposition: (tree \rightarrow include \rightarrow binary tree). In several maps an additional node has been inserted (tree \rightarrow has \rightarrow types \rightarrow includes \rightarrow binary tree). These intermediate nodes frequently labeled words like types, examples, terminology, or comparison. Rather than introducing new concepts, these nodes clarify the relationships between other concepts. It may be useful to compile a list of these words and use that list to automatically adjust the map representation provided to the flooding algorithm.

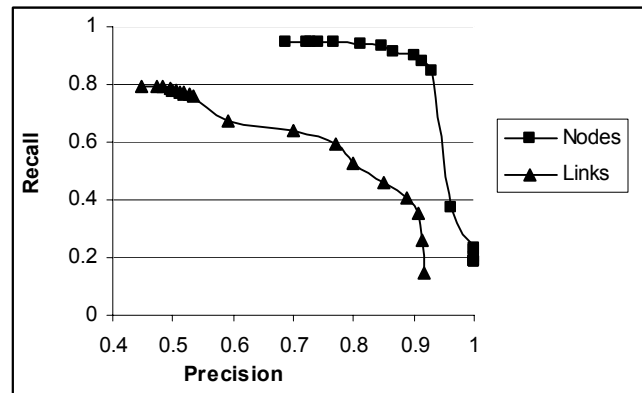


Fig. 11. Recall vs. Precision

8 Conclusions and Future Directions

The combined structural and semantic matching presented in this work is a promising approach for matching concept map elements. This work identifies a need for element level matching in concept maps and it explores the use of schema matching techniques. Existing concept map evaluation techniques were reviewed for common themes and measures. Existing schema matching algorithms were reviewed in a concept map matching context. A candidate algorithm, the similarity flooding algorithm developed by Sergy Melnik, Hector Garcia-Molina, and Erhard Rahm, was identified and tested to establish a performance baseline for future work. The system improved on simple string match results but employed only readily available information such as common abbreviations, key terms, and node colors.

Previous computerized concept map applications measure overall map similarity but do not emphasize the element matches needed in providing student feedback. Previous educational and knowledge management concept map application research has generally considered only conceptual nodes and not propositional links [3, 5]. Our proposed approach identifies these links as a computationally important dimension of the knowledge contained in a concept map. Also, in contrast to previous concept map algorithms, our system uses concept maps as they are generally created in educational settings without controlling the list of potential nodes or adding importance

weights to the maps. Even so, in our experimentation with student-drawn maps, we were able to identify 91% of the correct node and proposition matches. The assumptions made in the design of many database-oriented semantic-integration systems may not be all that appropriate for more informal concept-graph matching applications, as discussed in Section 2.3 above. Because human-drawn concept maps are similar to other node-link knowledge structures such as semantic webs, ontologies, work flow descriptions, and web service implementation models, improved concept mapping techniques may also have important non-education applications.

To guide improvements to the matching process, we identified a few commonly occurring organizational variations and noted their relationship to educational and cognitive processes. We tested both simulated and human drawn concept maps exhibiting these variations. Because these variations negatively impacted matching accuracy, the system was adjusted in several context-appropriate ways to increase accuracy. We introduced “node anchoring” to lock in key terms and automatically recognized some important hierarchical clusters using node colors, link names, and link directions. Using a matching algorithm such as the one described in this work, educational map evaluation and feedback processes might be improved. Mapping suggestions might also be provided for students by leveraging a collection of existing maps. It is hoped that this kind of prompting or tutoring will have a positive effect on learning and knowledge acquisition.

We plan to implement element matching in a semi-automatic scoring system and measure its impact on student feedback processes. Element matching is needed for such a system. Establishing a mapping between single nodes and between concept→link→concept substructures is a good beginning for matching larger structures such as hierarchical clusters. We plan to augment our mapping system to leverage the SF multimapping to identify some of these larger substructure matches. Although our current implementation individually matches a query map to a target map, information gathered in matching one map may be useful in matching other maps in the same collection. We plan to explore this possibility in the system as it is developed. Finally, because student misconceptions are educationally important and have a negative effect on matching accuracy, we intend to add misconception detection capabilities to the concept map evaluation system.

9 Acknowledgements

We would like to thank the NSF for supporting this project. NSF National STEM Education Digital Library: “Intelligent Collection Services for and about Educators and Students: Logging, Spidering, Analysis and Visualization” Award No. DUE-0121741, Program 7444. September, 2001-August 2003. We also would like to thank the GetSmart team and other members of the U of A’s AI Lab who built GetSmart components, especially Benjamin Smith, Chun Q. Yin, and Steven Trush. Finally, we recognize and appreciate the efforts of Ed Fox, Rao Shen, and Lillian Cassel in evaluating

the GetSmart system and providing important feedback guiding its development.



Byron Marshall is an Assistant Professor of Information Management at Oregon State University. He received a Ph.D. in Management Information Systems from the University of Arizona in May, 2005. His research interests emphasize the re-use of organizational data in informal node-link knowledge representations to support analysis tasks. His work includes applications in bioinformatics, business intelligence, digital library, law enforcement, and education.



Hsinchun Chen is the McClelland Professor of Management Information Systems (MIS) at the Eller College of the University of Arizona. He received his Ph.D. degree in Information Systems from New York University. He is author of more than 200 articles covering medical informatics, knowledge management, homeland security, semantic retrieval, and Web computing in leading information technology publications. He is a scientific counselor/advisor of the Lister Hill Center of the National Library of Medicine (NLM/USA) and the National Library of China. Dr. Chen is the director of the University of Arizona’s Artificial Intelligence Lab. <http://ai.bpa.arizona.edu/hchen>



Therani Madhusudan (Madhu) is currently VP of Engineering and Product Development at SITSCORP Inc, CA., leading the development of business process automation tools. He was an Assistant Professor at the MIS Department, University of Arizona from Fall 2000 to Summer 2005. He holds Ph. D. (1998) and M. S. degrees (1994) from Carnegie-Mellon University and a B.Tech (1990) from the Indian Institute of Technology, Madras, India. He has published over 25 refereed research articles in academic conferences and journals in the areas of Workflow Management, Information Integration, Product Lifecycle Management, and Engineering design automation.

10 References

- [1] A. J. Canas, J. W. Coffey, T. Reichherzer, G. Hill, N. Suri, R. Carff, T. Mitrovich, and D. Eberle, A performance support system with embedded training for electronics technicians, presented at The Proceedings of the Eleventh Annual Florida Artificial Intelligence Research Society Conference, 1998.
- [2] A. J. Canas, D. B. Leake, and D. C. Wilson, Managing, mapping and manipulating conceptual knowledge: exploring

the synergies of knowledge management & case-based reasoning, presented at AAAI Workshop on Exploring Synergies of Knowledge Management and Case-based Reasoning, Menlo Park, CA, 1999.

[3] A. J. Canas, D. B. Leake, and Maguitman, Combining concept mapping with CBR: Experience-based support for knowledge modeling, presented at Fourteenth International Florida Artificial Intelligence Research Society Conference, 2001.

[4] M. J. Carnot, B. Dunn, A. J. Canas, P. Gram, and J. Muldoon, Concept maps vs. web pages for information searching and browsing, vol. 2002: Institute for Human and Machine Cognition, 2001.

[5] S.-W. Chen, S. C. Lin, and K. E. Chang, Attributed concept maps: fuzzy integration and fuzzy matching, IEEE Transactions on Systems, Man, and Cybernetics, 31(5) (2001).

[6] T. L. Chmielewski and D. F. Dansereau, Enhancing the recall of text: Knowledge mapping training promotes implicit transfer, Journal of Educational Psychology, 90(3) (1998).

[7] A. Doan, Y. Lu, Y. Lee, and J. Han, Object Matching for Information Integration: A Profile-Based Approach, IEEE Intelligent Systems, Special Issue on Information Integration on the Web (2003 to appear).

[8] A. Doan and A. Halevy, Semantic-Integration Research in the Database Community, AI Magazine, 26(1) (2005).

[9] K. Ford, J. W. Coffey, A. J. Canas, E. J. Andrews, and C. W. Turner, Diagnosis and explanation by a nuclear cardiology expert system, International Journal of Expert Systems, 9 (1996).

[10] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais, The Vocabulary Problem in Human-System Communication, Communications of the ACM, 30(11) (1987).

[11] B. R. Gaines, Class library implementation of an open architecture knowledge support system, International Journal of Human-Computer Studies, 41(1-2) (1995).

[12] B. R. Gaines and M. L. G. Shaw, Concept maps as hypermedia components, International Journal of Human-Computer Studies, 43(3) (1995).

[13] B. R. Gaines and M. L. G. Shaw, WebMap: concept mapping on the Web, presented at Proceedings of WWW4: Fourth International World Wide Web Conference, Boston, 1995.

[14] R. R. Hoffman, J. W. Coffey, K. Ford, and M. J. Carnot, Storm-1k: A human-centered knowledge model for weather forecasting, presented at The Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society, Minneapolis, MN, 2001.

[15] I. M. Kinchin, If concept mapping is so helpful to learning biology, why aren't we all doing it?, International Journal of Science Education, 23(12) (2001).

[16] R. Kremer, Concept mapping: informal to formal, presented at Proceedings of the International Conference on Conceptual Structures, University of Maryland, 1994.

[17] J. G. Lambiotte, D. F. Dansereau, D. R. Cross, and S. B. Reynolds, Multirelational semantic maps, Educational Psychology Review, 1 (1989).

[18] D. B. Leake, A. Maguitman, and A. J. Canas, Assessing conceptual similarity to support concept mapping, presented at Proceedings of FLAIRS-02, Pensacola, Florida, 2002.

[19] B. Marshall, Y. Zhang, H. Chen, A. Lally, R. Shen, E. Fox, and L. Cassel, Convergence of Knowledge Management and E-Learning: the GetSmart Experience, presented at Joint Conference on Digital Libraries, Houston, Texas, 2003.

[20] S. Melnik, H. Garcia-Molina, and E. Rahm, Similarity flooding: a versatile graph matching algorithm (extended technical report), vol. 2003, 2001.

[21] S. Melnik, H. Garcia-Molina, and E. Rahm, Similarity flooding: a versatile graph matching algorithm and its application to schema matching, presented at Proceedings of the 18th International Conference on Data Engineering (ICDE '02), San Jose, Ca, 2002.

[22] J. T. Nosek and I. Roth, A comparison of formal knowledge representation schemes as communication tools: predicate knowledge vs. semantic network, International Journal of Man-Machine Studies, 33 (1990).

[23] J. Novak and D. B. Gowin, Learning How To Learn. (Cambridge, UK: Cambridge University Press, 1984).

[24] J. Novak, Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations. (Mahwah, New Jersey: Lawrence Erlbaum Associates, 1998).

[25] N. F. Noy and M. A. Musen, Prompt: Algorithm and tool for automated ontology merging and alignment, presented at Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, USA (2000), 2000.

[26] E. Rahm and P. A. Bernstein, A survey of approaches to automatic schema matching, The VLDB Journal, 10 (2001).

[27] J. A. Rye and P. A. Rubba, Scoring Concept Maps: an Expert Map-Based Scheme for Relationships, School and Science Mathematics, 102(1) (2002).

[28] R. J. Shavelson, H. Lang, and B. Lewin, "On concept maps as potential "authentic assessments" in science.," National Center for Research on Evaluation, Standards, and Student Testing. US Department of Education, Grant R117G10027 (ERIC Document Reproduction Service No. ED 367691), Los Angeles, CA 1993.

[29] M. Weideman and W. Kritzinger, Concept mapping vs. web page hyperlinks as an information retrieval interface: preferences of postgraduate culturally diverse learners, presented at Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on enablement through technology, 2003.