Experience-based support for human-centered knowledge modeling

David Leake[a], Ana Maguitman[b], Thomas Reichherzer[c]

[a]*School of Informatics and Computing - Bloomington, Indiana University, Informatics West, 901 E. 10th St., Bloomington, IN 47408, USA*
[b]*CONICET and Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Avda Alem 1253, Bahía Blanca B8000CPB, Argentina*
[c]*Department of Computer Science, University of West Florida, Bldg. 4, 11000 University Pkwy., Pensacola, FL 32514, USA*

Abstract

The construction, capture and sharing of human knowledge is one of the fundamental problems of human-centered computing. Electronic concept maps have proven to be a useful vehicle for building knowledge models. However, the user has to deal with the difficult task of deciding what information to include in these models. This article reports the culmination of a multi-year research project aimed at developing intelligent suggesters designed to aid users of concept mapping tools as they build their knowledge models. It describes DISCERNER and EXTENDER, two proactive suggesters that can be incorporated into the CmapTools concepts mapping system. DISCERNER applies case-based reasoning techniques to suggest potentially useful propositions mined from other users' knowledge models, while EXTENDER mines search engines to suggest new related areas to model. The article presents experimental results addressing two previously open questions for the project: DISCERNER'S retrieval accuracy and EXTENDER'S ability to generate artificial topics with content similar to topics determined by domain experts. Both experiments show satisfactory results.

Keywords: Case-based reasoning, Concept mapping, Intelligent user interfaces, Knowledge modeling, Knowledge discovery

1. Introduction

Human-centered computing (HCC) (e.g., [21, 43, 3]) studies methods for improving the interactions and performance of combined human/machine systems. A key challenge for human-centered computing is how to facilitate the construction, capture, and sharing of human knowledge. The knowledge-based systems community is well aware of the difficulty and cost of building knowledge models, which has led to interest in leveraging experience to aid knowledge modeling. This article presents research on applying ideas from case-based reasoning (CBR) (e.g., [34]) to the task of knowledge modeling, supporting users of software tools for concept mapping. Concept mapping [38] aims to elucidate a particular individual's conceptualizations about a domain, putting them in an explicit form which can be compared. It has proven a useful approach for constructing and sharing knowledge without requiring formalization, enabling end users to capture knowledge with minimal training. However, users faced with the task of developing a concept map may not always be able to remember all the most relevant concepts, or may have difficulty deciding which concepts to add to a concept map under construction (referred to as *extending* the concept map). Likewise, it may be difficult for users modeling a domain to identify the topics for which concept maps should be generated. Consequently, there is need for tools to support the concept mapping process.

The article describes research on the development of "intelligent suggesters" designed to proactively provide information to aid users of knowledge model-

E-mail addresses: leake@cs.indiana.edu (D. Leake), agm@cs.uns.edu.ar (A. Maguitman), treichherzer@uwf.edu (T. Reichherzer).

Please cite this article in press as: David Leake, Ana Maguitman and Thomas Reichherzer. Experience-based support for human-centered knowledge modeling, Knowledge-Based Systems (2014), http://dx.doi.org/10.1016/j.knosys.2014.01.013

ing tools for concept map construction. It describes DISCERNER[1], an experience-based system which aids knowledge modelers by drawing on other users' knowledge models, presenting suggestions mined from them, and EXTENDER[2], a system which complements DISCERNER's experience-based approach by drawing on information mined from search engines to help identify novel connections to consider and new areas to model. Both systems are optional software components which can be incorporated into the CmapTools [9] concept mapping system to augment its functionality. Together they provide context-relevant support both for leveraging the knowledge in prior concept maps and for going beyond the knowledge prior maps contain.

This article begins by discussing the task context for this work—the problem of supporting concept-map-based knowledge modeling—and the opportunity for knowledge extension support using ideas from CBR. It then presents the methods that we have developed for knowledge extension and topic generation. The methods developed have been evaluated individually in controlled experiments, as well as informally tested as robust prototypes within CmapTools [9], a widely-used knowledge modeling system developed at the Institute for Human and Machine Cognition (IHMC), with encouraging results. This paper reports the culmination of a strand of research begun in 2002 in collaboration with Alberto Cañas and the CmapTools team (e.g., [25, 29, 32, 28]). As the first journal publication on our work on DISCERNER and EXTENDER, the article summarizes key ideas; it also presents new results of experiments designed to address key open questions remaining from previous publications: The accuracy of DISCERNER'S indexing and the ability of EXTENDER to generate topics similar to those generated by human experts. The article closes with a review of related work, addressing the ramifications of this work for CBR and knowledge capture interfaces.

2. Supporting knowledge modeling with concept maps

Concept mapping [38] was first proposed in education, to enable students to externalize their knowledge by constructing a two-dimensional, visually-based representation of concepts and their relationships. This representation was seen as elucidating their internal cognitive structures, suitable for assessment or knowledge sharing. Concept mapping is used worldwide to facilitate knowledge examination, construction, comparison, and reuse by users ranging from elementary school students to scientists (for a recent sampling of its use, see [11]).

The CmapTools software [9] supports generation, storage of, and access to concept maps in electronic form. In addition to providing basic operations needed to draw and label concept maps, CmapTools includes extensive capabilities for annotating concept maps with links to electronic resources such as images, diagrams, video clips, and other concept maps, enabling the construction of richly connected concept-map-based *knowledge models* for particular domains. It also enables distributed storage and access to concept maps on multiple servers, to support knowledge sharing across multiple sites. Figure 1 shows a screen image captured during a session in which a user was extending a concept map. The bottom left window shows a sample concept map of the Mars exploration domain; the window for the initial concept map is the starting window for the system. The next window to the right is the window opened for DISCERNER and EXTENDER's suggestions when the user invokes those systems. The top portion of the suggestion panel presents a list of propositions suggested by DISCERNER, and the bottom of this panel presents topics suggested by EXTENDER. We describe these windows and their use further in Section 3. The windows arranged on the border of the image were generated by the user during the CmapTools session, by clicking on icons associated with nodes of the concept map in the starting window. In this instance, the new windows contain (clockwise from the original concept map window) an image, a related concept map, and a web page.

Many systems have been developed to facilitate human capture of knowledge in formal representations suitable for machine reasoning; for example, an extensive set of ontology editing tools has been developed (for reasons of space, we cannot summarize them here; see Denny [18] for a survey). The CmapTools project contrasts in taking a human-centered view, aiming to support the capture of knowledge in a form conducive to human examination and sharing. Concept maps provide an "informal," nonstandardized representation based on structured, simplified natural language. Electronic concept mapping has been successfully applied to knowledge capture and sharing for a wide range of tasks such as maintaining Navy equipment [10], local weather prediction [20], explaining the design of rocket engines [14], and describing Web services in Service-Oriented Architectures (SOA) composite applications [13]. An overview of a number of applications of concept map-

[1] Decision Index for Searching Category Entries by Reducing NEighborhood Radius.

[2] EXtensive Topic Extender from New Data Exploring Relationships.

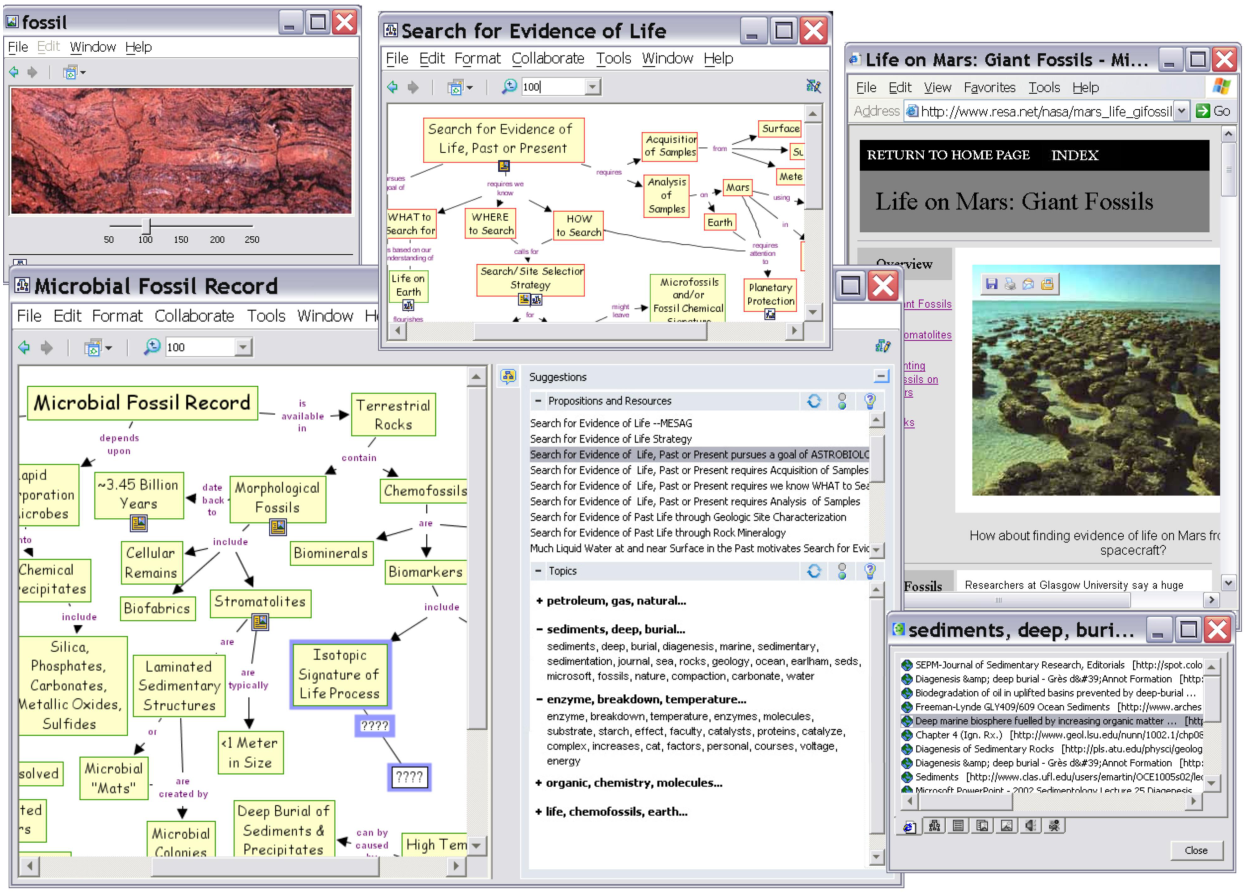

Figure 1: The CmapTools interface. The figure illustrates a concept map under construction, associate resources, and suggestions for concept extensions.

ping tools is presented in [36].

Informal studies show that when building concept maps, both experts and ordinary users often pause for significant amounts of time wondering what information to include. Frequently, they look at existing concept map libraries and information on the Web for concepts or links to include in their maps or for topics to start new maps for creating rich, comprehensive knowledge models. Our tools aim to automatically provide suggestions generated from such sources.

3. Using case-based reasoning to support knowledge modeling

During knowledge modeling, concept maps are constructed incrementally. At each step, concepts in an in-progress concept map are "extended" by adding new connections, either to existing or new concepts. Thus the user chooses a concept of a partial concept map, in context of already-existing concepts and links, and selects an appropriate link or concept/link pair to add to the map, connected to the chosen concept. If previous users have confronted similar situations when building concept maps, and have resolved them with particular choices, that knowledge modeling experience may be reused. Such reuse fits within the mold of case-based reasoning (CBR), which solves new problems by retrieving and adapting the solutions of similar prior problems (e.g., [34]). DISCERNER's task is to propose concept map extensions to the user. Because the concept mapping process aims to capture the current user's conceptualizations, there is no single "right" extension; the system plays an advisory role.

For example, suppose the user is an astrobiologist building a knowledge model composed of concept maps on the Mars domain, as in Figure 1, and that the user's current task is to construct a concept map on "Microbial Fossil Records." The user may require additional material to include in the in-progress concept map, as well as suggestions of related topics to begin the construction of new concept maps for the Mars knowledge model.

The top area of the suggestion panel contains DISCERNER's suggestions, based on prior concept maps, which may include concepts, propositions (pairs of linked concepts) and other resources. In the screenshot these include the proposition "Search for Evidence of Life, Past or Present, pursues a goal of ASTROBIOLOGY." When the user selects this proposition (by clicking), the system opens the concept map in which it appears, "Search for Evidence of Life" (shown on the top center of Figure 1), providing rich related material to consider for inclusion in the concept map being built.

Simultaneously, EXTENDER generates suggestions of candidate topics related to, but distinct from, the concept map under construction; these are potential topics for additional concept maps to generate. For each topic, it generates a top-level label consisting of three terms, used to characterize the suggested topic. Any of these suggestions can be expanded by clicking on the "+" symbol appearing at the left side of the topic label, to show additional terms for the topic. In our example, suggestions "sediments, deep, burial" and "enzyme, breakdown, temperature" have been selected for expansion and additional terms are displayed. Furthermore, the user can examine the web pages that were used by EXTENDER as a starting point for generating its suggested topics, as shown at the bottom-right of Figure 1, presenting a list of web pages associated with the topic "sediments, deep, burial".

Applying CBR to extending concept maps presents challenges. First, CBR generally treats cases as being segmented into problem–solution pairs. The problem part is used to identify relevant prior cases (those whose problem parts are similar); once a relevant prior case is found, its solution is adapted to solve the new problem. When cases represent concept maps, there are no static "problems" and "solutions"; the system must retrieve prior maps with some subpart relevant to whatever subpart of the user's new map on which the user is currently working. Consequently, it is not possible to pre-define the "problems" and "solutions" for concept map cases; these parts depend dynamically on the area of the new concept map being extended.

Second, effective CBR depends on efficient access to stored cases, which is often achieved by indexing stored cases according to indices drawn from carefully crafted and standardized "indexing vocabularies." Concept maps have nonstandardized representations—different users may label the same concept or link differently. Consequently, indexing and retrieval of concept map cases must be able to find good cases without strong assumptions of representational uniformity. Likewise, for scalability (to efficiently retrieve prior concept maps from potentially extensive libraries of concept maps, on any topic the user may propose), tools to aid concept map extension must include domain-independent methods for automatically generating indices from concept maps. In needing to retrieve prior cases which have not been pre-structured, and in having to deal with unrestricted vocabularies, the concept map suggester task faces some of the same issues faced by textual case-based reasoning [47]. However, concept maps' explicit links between concepts provide a valuable additional information source beyond what is available in text. A major focus of our work is on developing methods to exploit this information.

4. Mining concept map libraries for cases to support knowledge extension

DISCERNER helps a user to create extensions for a concept by linking it to other concepts in the same map or new concepts added to the map. For example, consider the scenario in Figure 1, for which DISCERNER is aiding in connecting concepts to "Search for Evidence of Life, Past or Present." First, DISCERNER retrieves similar prior concept maps that include "Search for Evidence of Life, Past or Present" or concepts with similar textual descriptions. After this set of candidate concept maps has been retrieved, the system extracts the ways "Search for Evidence of Life" was linked to other concepts in those past contexts and lists them in a side panel as candidates for potential extensions of a highlighted concept in the map.

Consistent with our observation that any part of the concept map may be seen as the "problem"—the concept to extend by adding links to other concepts—or the solution—the concepts to be added to the concept map and/or linked, DISCERNER's indexing approach does not pre-define "problems" or "solutions." Instead, it classifies concept maps into a hierarchical set of categories providing a broad characterization of the material in the map, and bases retrieval on that characterization. For efficient matching/retrieval it uses a vector space model [41] to describe the content of each category and each concept map. Terms in the vector correspond to terms appearing in the concept and link labels in the map. Because structural information plays an important role in determining the topic of a concept map, one of the challenges for this work has been to reflect structural information in the vector space representation.

4.1. Reflecting structure in a vector space model

Clustering of concept maps requires a similarity measure to approximate the semantic similarity between concept maps. For weighting terms in the vector space model, the term-frequency - inverse document frequency (TF-IDF) [41] approach is often used. TF-IDF adjusts the frequency of a term in a document by an inverse frequency of the same term's occurrence in the document library, and constructs term vectors in which each term is weighted by its TF-IDF value. The resulting vector can then be used for document comparison. In principle, TF-IDF could be applied to the terms appearing in concept and link labels of a concept map to develop a term vector representation for it, simply treating the terms in the concept map labels as text. However, that approach is inadequate for two reasons. First, it ignores valuable information contained in the concept map's structure. Second, term frequencies are not useful for concept maps: in a well-designed concept map, terms are seldom repeated, regardless of their importance. DISCERNER's approach is based on assessing the importance of concepts—and hence, of the terms in the concepts' labels—based on the topological structure of a concept map, and weighting them according to that importance. The method for assessing importance is based on human-subjects studies of how concept map structure affects human judgments of concept importance, as described below.

Modeling structure-based influences on human concept importance judgments. In order to present users with suggestions that reflect the concepts they tend to find most important in concept maps, DISCERNER's concept weightings are designed to reflect human concept importance judgments. We developed three candidate models of how structural factors affect concept importance ratings, and then performed human-subjects experiments to fit them to human concept importance judgments [27]. Some of the factors they consider are inspired by general guidelines for constructing "good" concept maps, taken from the concept mapping literature [38], e.g., reflecting the importance of concept maps' hierarchical structure by weighing upper and lower concepts differently. Others are inspired by methods for analyzing the topology of hyperlinked network structures [24], e.g., that nodes in a graph may be characterized based on the number of outgoing and incoming connections as either "hubs" or "authorities." A node in a graph structure is a hub node if it has many outgoing links, and an authority node if it has many incoming links, relative to other nodes in the graph. Outgoing and incoming links refer to the direction by which nodes are connected in a directed graph; we consider concepts as the nodes and connecting linking phrases as the links in a graph. Thus, "hub" concepts are concepts with many outgoing connections to other concepts (in the form of the links of propositions); in contrast, "authority" concepts have many incoming connections from other concepts. Hub concepts tend to appear at the beginning of propositions, while authority concepts tend to appear at the end of propositions. We hypothesized that hub and authority characteristics of concepts might play a significant role in describing a map's content, while abstracting away from low-level structural detail.

The models are summarized in table 1. The structural influences that they consider include (1) distance of a concept to a root concept, measured in terms of the number of links on the shortest path to the root concept in the concept map graph, (2) connectivity of a concept measured in terms of the number of incoming and outgoing connections, and (3) the concept's global connectivity to the root concept measured by a "path frequency" (PF) measure. In addition, models that consider multiple influences have parameters to weight the different influences.

For the "connectivity root-distance" (CRD) measure, the model parameters α, β, and δ adjust the effect of the number of incoming connections $(i(c))$, the number of outgoing connections $(o(c))$ and the distance to the root concept $(d(c))$ of a concept c. For the "hub-authority-root-distance" (HARD) measure, the model parameters α, β, and γ adjust the effect of the authority $(a(c))$, hub $(h(c))$ and upper concept $(u(c))$ (concepts appearing near the top of the concept map) value of c. These values correspond to the concept's role as an authority, hub, and upper concept, while the upper weight reflects proximity of a concept to the root concept (for full details, see [26]). PF counts the number of distinct paths that reach a given concept c, traversing the paths of the concept map graph starting from the root concept $(n(c))$, and requires no parameters.

To our knowledge, no previous studies had explored the role of such factors in human judgments of concept importance. We conducted a set of experiments involving 20 participants selected from students and staff at Indiana University, using concept maps specifically designed to investigate structural influences by varying factors such as the degree of connectivity or the distance to the root concept of certain concepts in the map. The maps' concept and link labels were replaced with random letter combinations, to observe structural influences independently of map content. Participants were presented with pairs of concepts and asked to se-

Table 1: Models for assessing concept importance.

Connectivity Root Distance (**CRD**)	$w(c) = (\alpha \cdot o(c) + \beta \cdot i(c)) \cdot (1/(d(c)+1))^{1/\delta}, \qquad \alpha, \beta >= 0, \delta >= 1$
Hub Authority Root Distance (**HARD**)	$w(c) = (\alpha \cdot a(c) + \beta \cdot h(c) + \gamma \cdot u(c)),\ \alpha, \beta, \gamma >= 0$
Path Frequency (**PF**)	$w(c) = n(c)$

lect the more important one, or to indicate that both were equally important. We then fitted the models to the participants' preferences by adjusting the models' parameters using a hill-climbing algorithm. The results revealed two trends in structural effects on concept importance: (1) both authority concepts and concepts with incoming connections are considered more important than hub concepts or concepts with outgoing connections, and (2) concepts close to the root concept are considered more important than concepts more distant from the root concept [27]. These results enable us to model structural influences on concept importance and to choose appropriate model parameters for CRD and HARD to weigh the terms occurring in the concept labels.

Using structure-based weightings in similarity assessment. Given a selected model from table 1 and a concept map C, the term-vector representation of the concept map is derived by extracting terms k from the concepts in the map and assigning each term a weight defined as the sum of the weights $w(c_i)$ for all concepts c_i in the concept map in which k occurs, according to the selected model, normalized by the largest term weight in the concept map C. Similarity assessment between concept maps is then done by cosine similarity.

We note that our vector-space model takes the links between concepts into consideration for concept weighting, but that the linking phrases are not included in the representation of a concept map for constructing an index. We have conducted experiments whose results support the hypothesis that concepts are generally more valuable than linking phrases when building a representation for indexing concept maps [30]. The experiments involved generating Web queries from selected concepts and links in a concept map, submitting them to a search engine and comparing matching Web documents to the map, to determine which keywords are better search terms and therefore better indexing terms. While queries constructed from one or more concepts yield good results, those constructed from linking phrases returned documents with little similarity to the concept map from which they came. This indicates that concepts are generally more useful than linking phrases when building a representation for indexing concept maps. A detailed discussion of the experimental method and results can be found in Leake et al. [30].

4.2. Building an index from concept maps

In the context of Web search, indexing is often done by computing sets of hierarchical categories such that documents within a category are more closely related than the documents from different categories. We have developed domain-independent methods to automatically create such an index from the vector space representation of concept maps [26], aimed at being useful in any domain the user may choose. In contrast to the knowledge-rich—and often hand-generated—indices of many CBR systems, DISCERNER's index is generated automatically with no background knowledge.

The index organizes concept maps into a hierarchical tree structure of clusters, each containing a set of concept maps involving correlated concepts. More tightly coupled clusters of concept maps appear towards the bottom of the tree structure, and more loosely coupled clusters towards the top. For each cluster, the index maintains references to the original concept maps and keeps a cluster representative, to serve as a prototype for comparing clusters and to determine if a concept map is related to a set of clustered maps. The cluster representative is computed from the maps in the cluster as described below.

At the bottom of the hierarchical structure are the leaf nodes of the tree, each organizing a cluster of maps. This cluster forms the most specific category of maps in the library. Each leaf node in the tree may be subsumed by several parent nodes, that form larger clusters representing more generic categories. When a concept map library is indexed, each concept map is assigned to a single leaf node; all leaf nodes together form a partition of the concept map library. Each element of the partition can be seen as corresponding to a topic, as defined by shared vocabulary among the maps in a category. These do not resemble human-derived classifications of maps, but are useful in searching for related concept maps that serve as candidates for possible extensions to a map, as shown in the experiment in Section 4.5. The search process for related concept maps starts from the top of the tree with the most generic category of maps and continues downward in the tree. To identify similarity between a map and a category of maps, keywords from the cluster representative are compared against the concept map. Figure 2 illustrates DISCERNER's the process for finding concept maps similar in content to an input concept map.

4.3. DISCERNER*'s index generation algorithm*

DISCERNER uses an agglomerative algorithm to compile concept maps into a hierarchical, tree-like structure, as described in Algorithm PROCEDURE GENERATE-INDEX. The algorithm starts from a set of initial clusters each containing a single concept map, and then repeatedly merges the clusters whose cluster representatives are most similar to each other (by the metric described in section 4.1), making each merged cluster the parent of the clusters that were merged. Initially, when each cluster contains only a single map, that map is chosen as cluster representative. Subsequently, when two clusters are merged, a weighted sum of the cluster representatives from the clusters being merged becomes the cluster representative of the merged cluster. If r_1 and r_2 are the representatives of clusters 1 and 2, with cluster sizes n_1 and n_2, the sum is $(n_1 * r_1 + n_2 * r_2)/(n_1 + n_2)$. The merge process is continued until all clusters have been merged or the similarities between the cluster representatives fall below a pre-set threshold, suggesting that the concept maps from different clusters have little in common and should remain distinct.

During the merging process, the algorithm derives a new tree structure by creating categories from merged clusters and linking them to each other, with categories created according to three rules: The first rule creates a category from any cluster that has reached a pre-set minimum size. The second rule creates a category from any cluster that is the product of a merge of two clusters above the minimum size. For this rule, categories are generated from the individual clusters and from the merged cluster, with the categories of the individual clusters becoming a subcategory of the merged cluster. The third rule creates a category from a cluster if the corresponding subcategory and the new category are sufficiently different. The rules affect the depth and the width of the category index's hierarchical structure, creating categories from clusters that are sufficiently different from each other and ultimately reducing the storage requirements, as well as reducing the number of comparisons needed during retrieval to find the category of a new concept map.

Parameters were chosen based on experiments with sample concept maps, to minimize classification errors. Details are given in the experimental section below.

The vector-space model, the cosine-coefficient measure, and the agglomerative clustering algorithm ensure that maps similar in content are grouped together forming an increasingly more general group of concept maps, starting from the base of the hierarchy tree up to the root.

We envision that, for an application of this approach in CmapTools, indices and case libraries would be compiled periodically by the individual concept map servers and then uploaded to a designated index server. This server would be responsible for merging the different indices into a combined index and responding to queries from clients for relevant suggestions. The combined index could include several disjoint category hierarchies if the individual hierarchies are dissimilar.

4.4. Retrieving and ranking suggestions

DISCERNER's users can actively initiate search for suggestions by selecting the concepts they seek to extend, or can have the system monitor the concepts being added to the concept map and proactively make suggestions related to the most recent additions. Whether in user-driven or proactive mode, DISCERNER generates suggestions by generating a term vector representation of the the current concept map and extracting keywords from the concepts selected by the user or the suggester. Together, the keywords in the labels of the selected concepts and the vector representation form a query, processed locally by the client and remotely by a designated index server for concept maps published by other users. The retrieval algorithm uses the vector to perform a binary search for the best-fitting category, down the hierarchies in the combined index. By adjusting a slider, users can control how far the retrieval algorithm descends in the hierarchy tree to search for related concept maps. The further it descends, the fewer maps it finds, but those found are more closely related to the map in progress. Once a concept map has been selected for retrieval, the keywords from the selected concept labels are used to look up specific cases within the concept map. Within concept maps, propositions are generally represented by concept–link–concept triples (in rare cases, propositions extend over more than two concepts, requiring additional link–concept pairs). Propositions from the retrieved map whose initial concept labels

Algorithm PROCEDURE GENERATE-INDEX

INPUT:
L: a library of concept maps.
min_S minimum cluster size
max_S maximum cluster size
min_{simRR}: minimum cosine similarity between two cluster representatives
min_{simRC}: minimum cosine similarity between a cluster representative
and the representative of the corresponding subcategories

OUTPUT:
A tree-based category index for searching related concept maps.
BEGIN
// compute initial clusters
For each concept map c_i in L do:
(1) Use the concept map vector-space model of c_i as the cluster representative r_i.
(2) Make r_i the single element in a new cluster.
// merge clusters agglomeratively and generate categories
Repeat the following steps:
(1) Find cluster representatives r_i and r_j closest to each other using cosine similarity.
(2) If distance of r_i and $r_j < min_{simRR}$ do:
(2a) Unless existing, construct a category for each cluster corresponding to r_i and r_j.
(2b) Exit the loop.
(3) Merge clusters corresponding to r_i and r_j.
(4) Compute a new cluster representative r_{merged} for the merged cluster.
(5) Generate a new category for the merged cluster if:
(5a) The merged cluster r_{merged} exceeds a threshold min_S.
(5b) The merged cluster is a product of two large clusters whose size is greater
than max_S; generate categories for r_i and r_j if not existing and
make them a subcategory of the new category.
(5c) The merged cluster is sufficiently different from the corresponding subcategories
using min_{simRC} as criteria applied to r_{merged} and the cluster representative
of the subcategories.
END

have high keyword similarity with the labels of selected concepts are suitable candidate suggestions.

Retrieved concept maps are ranked based on a comparison of their keyword correlations to the target map, using a correlation metric described in [26]. Figure 2 summarizes the entire process of generating suggestions.

4.5. Evaluation of indexing performance

An open question from previous work on this project was the quality of retrievals based on DISCERNER's indexing process. This section presents new results addressing that question.

The effectiveness of retrieval methods depends on finding a sufficient portion of the relevant information, which is commonly measured by recall and precision, with recall measuring the fraction of relevant documents that have been retrieved and precision measuring the fraction of retrieved documents that are relevant. To assess the algorithms for generating category indices and compiling case libraries, several tests were conducted on two data sets. The first set contained three knowledge models on overlapping topics, respectively comprising 93 concept maps from the Mars 2001 library [6], 9 concept maps on the NASA Centaur Rocket System [14], and 14 maps on a meteorology project [20]. The second data set contained two knowledge models on dissimilar topics, with 14 maps on AI topics and 17 concept maps on water and glaciers. The experiment was designed to investigate, (1) whether similar topics are merged into a single hierarchy of categories while dissimilar topics would be kept separate, and (2) whether the generated index places the indexed maps in their assigned category, so that the recall of related maps and the precision of the retrieved maps are high (this testing strategy is related to the *leave-one-in* tests of [1]).

Table 2 summarizes the results from the experiment. For both data sets, we tested different input parameters of DISCERNER's category index algorithm, resulting in different index structures. The second column of the table shows the resulting number of leaf categories (i.e., partitions) of the concept map library. The third column shows a classification error, which is computed as the percentage of maps that could not be located by the retrieval process after the maps have been fully indexed.

The error of classification occurs because the retrieval process compares vector-representations of the map to the cluster representative that captures the theme of the maps in a cluster and not the individual maps. If a map is only marginally represented by the representative of a category an error may occur. Regardless of the different parameter settings for clustering, the algorithm

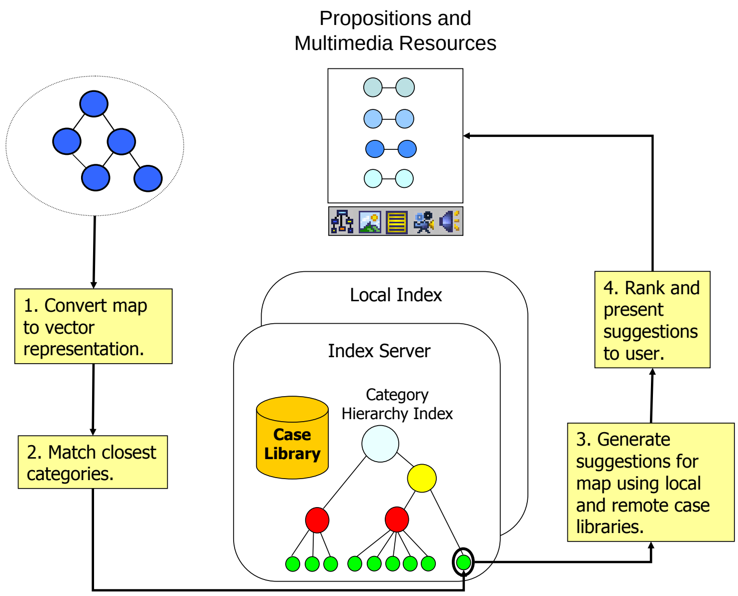

Figure 2: Generating extensions for a selected concept from a library of concept maps.

Table 2: Results from an automatic categorization.

	Tests	# Partitions	Classification Error
First	1	12	2.57%
Data Set	2	6	0.86%
	3	5	1.72%
Second	1	9	0%
Data Set	2	4	0%

computed a single category hierarchy for the first data set, and two separate hierarchies for the second. Thus the algorithm correctly determined that the models of the first set share common concepts, while the models of the second set have nothing or little in common.

For the maps of the first set that could not be located in the index, resulting in an erroneous classification, we examined the similarity of the maps with the maps they retrieved, and determined that the maps still related to the maps in the selected category. The similarity value ranged from .14 to .38 using the cosine measure when comparing a map with its best matching map in the category selected by the retrieval process (for identical representations, the value is 1.00). We also determined that the closest shared parent category in the hierarchy, subsuming both the incorrectly selected category and the correct category, is—except for one case—at most one step distant in the tree, meaning that the wrong category was selected in the final step of the lookup process. This is encouraging for the performance of the retrieval system, because this means that (in our tests) the original category would always be found if the user broadened the search to include a single additional level.

5. Mining search engines for additional information

Suggestions from previous concept maps are useful for elaborating new maps, but cannot help to extend the knowledge model beyond information that has already been captured in the concept map libraries. Users who are oblivious to having overlooked a topic may not realize *when* they need to seek information or *what* information to seek. Consequently, it may be useful for a suggester to go beyond known user desires, automatically forming queries seeking novel information that might be of interest to the user. EXTENDER's goal is to serve as a memory augmentation aid for users, generating cues for topics that they may have overlooked.

These topic suggestions, in conjunction with the context of the current concept map, may prompt remindings of relevant concept extensions or new concept maps to add to the knowledge model. EXTENDER [28], serves this goal by mining search engines to identify novel topics that go beyond previously captured information. EXTENDER currently draws on the World Wide Web, but it could also be applied, for example, to documents shared within an organizational intranet, to aid sharing of knowledge resources on individual machines.

Because retrieving and processing large numbers of Web pages is costly, EXTENDER begins its process with an inexpensive *distillation phase*, in which a series of queries is submitted to a search engine and only the information that is readily available from the search results (e.g. title, "snippet" of text, URL) is used to identify useful terms. To identify such terms in the context of a concept map, we have developed a framework for analyzing terms' importance as descriptors and discriminators of a topic [33, 31]. A term is a good document descriptor if it occurs often in the document, while a term is a good discriminator of a document if it occurs in the document but rarely occurs in other documents of the corpus. This formulation of term descriptive and discriminating power is in the spirit of traditional IR schemes. However, searching the Web to identify topics requires addressing new questions for the formulation of descriptors and discriminators.

A first question is what should be considered to be a "topic". One way to represent topics is implicitly, as sets of similar documents. The similarity between documents can be computed using the cosine measure. Then it is possible to determine if a term is a good *topic descriptor* by analyzing if it occurs *often* in the context of a topic. In other words, the terms that occur more frequently in documents similar to the concept map under analysis will be considered good descriptors of the concept map's topic. On the other hand, a term is a *good discriminator of a topic* if it tends to occur only in documents similar to the given topic. Therefore, terms with high discriminating power are expected to occur in some documents similar to the concept map under analysis but they should seldom occur in other documents.

The higher-level notions of topic descriptors and discriminators, as opposed to document descriptors and discriminators, help to identify important terms at the higher-level of topic. Topic descriptors and discriminators are extracted dynamically, by mining search engines. Once the best topic descriptors and discriminators are identified, they are used as query terms in a *search phase* to search for additional material on the Web. To achieve coverage, novelty, and diversity EXTENDER generates queries at incremental distances from the set of terms that originated the request. The system uses a *curiosity mechanism* to diversify during initial stages and focus towards the end. This approach is in the spirit of techniques such as simulated annealing and reinforcement learning, in which a temperature factor is set initially to favor exploration, and then adjusted to favor exploitation. In EXTENDER's process for extending a topic **T**, new terms are collected during each iteration. To control growth in the number of terms, whether new terms are added to **T** is regulated by a curiosity decay parameter.

EXTENDER's strategy for preserving global coherence is to use a *search context* for filtering irrelevant information and to identify good topic descriptors and discriminators for guiding query formation and subsequent retrievals. The collected material is clustered to identify topics in the collection, and unimportant material is discarded. This process is repeated a number of times, with the stopping criterion depending on a user-selected limit on iterations.

5.1. Evaluating EXTENDER*'s topic generation*

To judge EXTENDER's performance against an objective standard, we performed an experiment to evaluate the similarity of its artificial topics to the content of "gold standard" topics in experts' hand-crafted concept maps. As the gold standard topics, we used the set of concept maps in the Mars 2001 knowledge model. This knowledge model was created by experts from NASA and contains more than a hundred concept maps, presenting an extensive coverage of topics in the field.

In our tests the top-level concept map from the knowledge model was used as the starting point (corresponding to the map under construction) and EXTENDER's topic extension algorithm was used to produce a collection of artificial topics, without access to any of the other maps in the knowledge model. As a baseline method for comparison we implemented a simple algorithm which constructs queries using all the concepts from the same concept map EXTENDER used as a starting point, submits them as queries to Google, and clusters the results to generate topics. For comparison purposes we implemented another algorithm based on the SMART pseudo-relevance feedback (PRF) method [7]. This method constructs the initial queries in the same way as the other two methods, but refines them based on the pseudo-relevance feedback provided by the top 10 retrieved results. Results are also clustered to generate topics. We used the Google Web API with special permission from Google to carry out our evaluations; non-commercial search engines, such as Faroo (http://www.faroo.com) or Yacy (http://yacy.net), could serve as effective alternatives.

We expected EXTENDER's mechanism to provide results with superior global coherence, novelty, and coverage than the other two methods for equal number of Web queries. An evaluation based on global coherence and coverage requires an operational definition of *topic relevance*. Here, we consider the expert-generated Mars 2001 topics as *target topics*, with the relevance of a system-generated topic measured by the accuracy with which a system-generated topic replicates an expert-generated topic. Note that the accuracy measure also provides an indication of topic quality, because its results depend on the similarity between EXTENDER's topics and the expert-generated set, which we expect to be of good quality for the domain.

Because novelty is one of our desiderata for topic generation, we want to favor strategies that produce relevant topics with a high number of novel terms. Assume that $R = \{r_1, \ldots, r_m\}$ is a target set of relevant topics and $A = \{a_1, \ldots, a_n\}$ is a set of topics generated by the topic-generation strategy under evaluation. Consider the set o, containing the terms of the originating topic, i.e., the knowledge model that is used as a starting point to search for topics. We propose a similarity measure reflecting the proportion of *novel terms* (terms not in the starting knowledge model) in a system-generated topic a_i that are also part of an r_j from a set of relevant topics:

$$\textbf{Similarity}^N(a_i, r_j, o) = \frac{|(a_i \cap r_j) - o|}{|(a_i \cup r_j) - o|}.$$

The accuracy function can be written in terms of this similarity function, to measure the precision with which a given topic replicates some topic in the given set, disregarding those terms that are in the starting knowledge model:

$$\textbf{Accuracy}^N(a_i, o, R) = \max_{r_j \in R} \textbf{Similarity}^N(a_i, r_j, o).$$

We use this accuracy function to define a measure of global coherence that accounts for novelty:

$$\textbf{Global\_Coherence}^N(o, A, R) = \frac{\sum_{a_i \in A} \textbf{Accuracy}^N(a_i, o, R)}{|A|}$$

The coverage measure can be stated as

$$\textbf{Coverage}^N(o, A, R) = \frac{\sum_{r_i \in R} \textbf{Accuracy}^N(r_i, o, A)}{|R|}.$$

Parameter Settings. EXTENDER's methods depend on parameters such as the number of iterations (generations of topics), the number of queries submitted from the source concept map and from each generated topic, the maximum number of topic descendants for each topic, the starting and stopping thresholds for curiosity mechanisms and the similarity threshold for merging topics. This results in a large parameter space. In practice, however, pragmatic concerns for the interface, such as the desire for rapid response and low memory use, suggest constraining some parameters. Accordingly, our tests limited the number of generations to 4, the number of queries from each topic to 20 for distillation and 10 for search, and the number of topic descendants at each stage to 8.

Experimental Results. We first analyzed the performance of EXTENDER as a function of the number of iterations. The test was performed for 1, 2, 3 and 4 iterations. For each number of iterations our evaluation involved 48 trials, with different settings for EXTENDER's parameters. We observed that in general three iterations were sufficient to generate a rich variety of topics, with the system response time kept below 20 seconds. A smaller number of iterations significantly decreases coverage of novel material, while it usually increases global coherence.

When comparing the performance of EXTENDER against the other two methods, we set the number of iterations for EXTENDER and the pseudo-relevance feedback method to 3 and the number of queries for the baseline to the total number of queries submitted by the other two methods. For each trial, the three method used the same similarity threshold and method for merging topics.

Figure 3 presents a comparison of the performance of EXTENDER's topic generation algorithm to the other methods in terms of global coherence and coverage. A particular setting corresponds to a trial and is represented by a point. The point's x-coordinate corresponds to the performance of the baseline method for that case, the y-coordinate corresponds to the performance of the pseudo-relevance feedback method, while the z-coordinate corresponds to the performance of EXTENDER. In Table 3 we present the mean confidence intervals resulting from computing the performance criterion functions for the three compared methods. EXTENDER's results show statistically significant improvements over the other two methods.

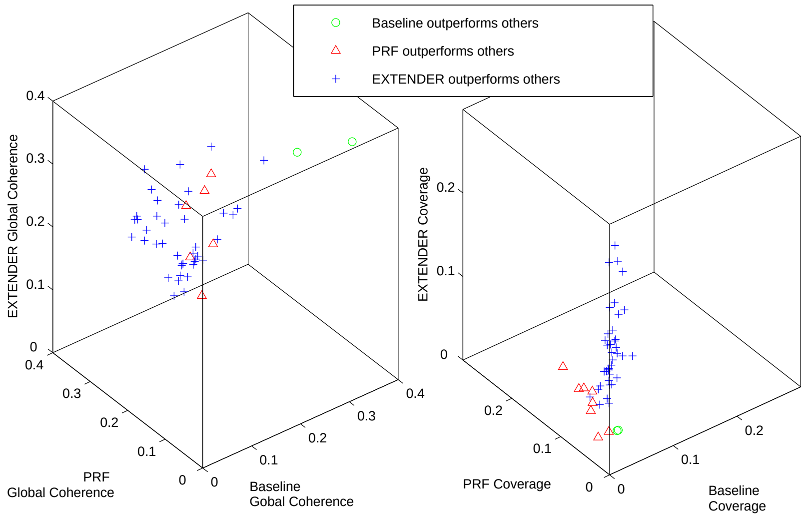

Figure 3: A comparison based on global coherence (left) and based on coverage (right).

Table 3: Confidence intervals for the mean global coherence (left) and mean coverage (right) of the three methods considering novel material only .

Method	N	MEAN	95% C.I.
EXTENDER	48	0.267	**(0.253, 0.281)**
PRF	48	0.168	**(0.149, 0.186)**
Baseline	48	0.101	**(0.077, 0.125)**

Method	N	MEAN	95% C.I.
EXTENDER	48	0.116	**(0.099, 0.132)**
PRF	48	0.029	**(0.023, 0.035)**
Baseline	48	0.019	**(0.017, 0.022)**

6. Related work

DISCERNER and EXTENDER's methods relate to a number of threads in CBR research. DISCERNER's extraction of cases from concept maps is in the spirit of previous systems which do data mining to extract cases from databases (e.g., [16, 40]); however, it contrasts in extracting cases on the fly, as needed, rather than in advance. This relates to research on dynamically extracting cases from domain knowledge [37], but because DISCERNER selects portions of predefined concept maps, its task is more constrained.

DISCERNER also contributes to the problem of assessing similarity of structured cases. Structural similarity assessment is an active research area in the process-oriented CBR community, which has focused primarily on making true structure matching more efficient (e.g., [4, 23]). In contrast, DISCERNER's approach uses an approximate structural summary. In domains for which automated reasoning will be applied to retrieved cases, full structural similarity is more important than in DISCERNER's support domain, for which the primary goal is to provide suggestions to jog a human user's memory, which can be done based on a subpart of a concept map.

A key issue for CBR-supported concept mapping is how to perform similarity assessment for non-standardized representations. Labels on concept map nodes provide names for the concepts that they represent, but not in the more formal, standardized representations assumed in much AI research (e.g., [46]). Node and link labels may be ambiguous or inconsistent with the names used in other concept maps. Such issues are a focus of research in textual CBR [47]. Our work focuses on methods for exploiting the additional structure provided by concept maps, and must also be robust to structural variations caused by non-standardized structures.

Similarity assessment is a core issue for case-based reasoning, with numerous approaches [15]. The primary contribution of DISCERNER's approach to similarity assessment is its use of structural summarization to enable highly efficient retrieval based on approximate structure, without costly structure mapping. EXTENDER replaces the standard task of "similarity assessment" with two-part retrieval criteria directly considering usefulness, which for EXTENDER depends on relevance and novelty, rather than similarity per se. Because a useful topic suggestion must go beyond the initial knowledge model, it must be *dissimilar* to the initial topic to some extent, though also related. The problem of measuring text-based diversity has also been studied [2]. However, EXTENDER not only addresses the problem of topic variety but also attempts to preserve coverage and global coherence.

The replacement of strict similarity with pragmatic considerations is in the spirit of adaptation-guided retrieval [44], and the need to span a set of alternatives relates to recent research on preserving case diversity during retrieval [45]. However, rather than focusing on how to *select* diverse alternatives, as in that work, EXTENDER focuses on how to *construct* a diverse set by incrementally searching similar items and tracing their divergent strands. The problem of extracting topics from streams of heterogeneous data has mostly been addressed in the context of social platforms [35].

Beyond CBR, the promise of the Semantic Web has prompted considerable interest in tools to aid the collaborative construction of ontologies (e.g., [17, 39]). For example, EXPECT [5] and SHAKEN [12], like CmapTools, aim to enable flexible knowledge acquisition without the mediation of knowledge engineers; SHAKEN is also based on a graphical interface. However, in contrast to concept-map-based approaches which retain informal knowledge, these systems' internal representations are based on formal languages. Other tools are aimed at reducing the bottleneck of knowledge acquisition in the construction of domain ontologies by applying text mining techniques. For example, Hsieh et al. [22] have used text mining techniques to support the extraction of concepts, instances, and relationships from a handbook of a specific domain to quickly construct a basic domain ontology. Reasons of space preclude an exhaustive summary, but other examples of semi-automatic construction of knowledge representation in the form of ontologies from existing data include work by Santos et al. [42] and by Gil and Martin-Bautista [19]. The systems described in this article are aimed at aiding the human reasoning process, based solely on knowledge captured in a human-friendly form. However, the methods described here could also be used to aid construction of formal representations. The Institute for Human and Machine Cognition is developing software tools to support rendering and editing Web ontologies, using concept maps to represent ontologies, with drawing conventions and transformations specifying precisely how an OWL (Web Ontology Language) ontology is mapped onto a concept-link graph structure and vice versa [8]. Once an ontology is represented as a concept map, the CmapTools interface and its suggesters can be applied to support ontology generation and extension.

The growing set of predefined standardized ontologies can help users to rapidly build their own ontologies by using existing and agreed-upon definitions of concepts, and we we see opportunities for applying the suggesters to supporting ontology extension, by retrieving new concepts and statements about concepts in the map from prior ontologies, and for suggesting new topics.

7. Conclusion

Electronic concept mapping tools provide a flexible framework for aiding knowledge capture and sharing, helping to empower experts to play an active role in the knowledge modeling process. Fully exploiting this framework requires supporting users as they perform the hardest part of concept map generation—selecting and relating the content to include. Because of the potential to reuse portions of prior knowledge models, representing others' concept-mapping experiences, CBR is a natural paradigm for providing such support. However, applying CBR ideas to concept mapping tools presents new challenges in areas such as efficiently assessing similarity of structured information and extracting cases from larger structures on the fly. Likewise, when prior knowledge models are insufficient, it may be necessary to complement CBR by going beyond captured experiences, and to draw on the Web as a whole, to mine cues to help the user's own process of remembering relevant information to add to the case library.

This article has presented research on methods for performing these tasks, implemented in DISCERNER and EXTENDER, including new experimental results on each system. These illustrate the promise of case-based approaches and web knowledge discovery to augment existing cases for human-centered knowledge modeling. In particular we have shown that the proposed tools are effective in retrieving suggestions from related knowledge models and that they significantly outperform other methods at recovering topics similar to those handcoded by an expert.

Acknowledgments

We gratefully acknowledge the contributions of our collaborators Alberto Cañas and the IHMC CmapTools team. This research is supported in part by NASA under award No NCC 2-1216. We thank Google for the use of their Web APIs service. Portions of this article are based on [29, 32, 28].

References

[1] Aha, D. and Breslow, L. (1997). Refining conversational case libraries. In *Proceedings of the Second International Conference on Case-Based Reasoning*, pages 267–278, Berlin. Springer Verlag.

[2] Bache, K., Newman, D., and Smyth, P. (2013). Text-based measures of document diversity. In *Proceedings of the 19th ACM SIGKDD (KDD '13)*, pages 23–31, New York, NY, USA. ACM.

[3] Bannon, L. (2011). Reimagining HCI: toward a more human-centered perspective. *interactions*, 18(4):50–57.

[4] Bergmann, R., Minor, M., Islam, M. S., Schumacher, P., and Stromer, A. (2012). Scaling similarity-based retrieval of semantic workflows. In *Proceedings of the ICCBR-12 Workshop on Process-Oriented Case-Based Reasoning*.

[5] Blythe, J., Kim, J., Ramachandran, S., and Gil, Y. (2001). An integrated environment for knowledge acquisition. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 13–20. ACM Press.

[6] Briggs, G., Shamma, D., Cañas, Carff, R., Scargle, J., and Novak, J. D. (2004). Concept maps applied to Mars exploration public outreach. In Cañas, A. J., Novak, J. D., and González, F., editors, *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*, pages 125–133, Pamplona, Spain. Universidad Pública de Navarra.

[7] Buckley, C., Singhal, A., and Mitra, M. (1995). New retrieval approaches using SMART: TREC 4. In *TREC*.

[8] C. Eskridge, T. and Hoffman, R. (2012). Ontology creation as a sensemaking activity. *IEEE Intelligent Systems*, 27(5):58–65.

[9] Cañas, A., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Arroyo, M., and Carvajal, R. (2004). CmapTools: A knowledge modeling and sharing environment. In *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain. University of Navarra Press.

[10] Cañas, A., Coffey, J., Reichherzer, T., Hill, G., Suri, N., Carff, R., Mitrovich, T., and Eberle, D. (1998). A performance support system with embedded training for electronics technicians. In *Proceedings of the Eleventh Annual Florida Artificial Intelligence Research Symposium*, pages 79–83. AAAI Press.

[11] Cañas, A. J., Novak, J. D., and Vanhear, J. (2012). *Concept Maps: Theory, Methodology, Technology. Proceedings of the Fifth International Conference on Concept Mapping*. Veritas Press.

[12] Clark, P., Thompson, J., Barker, K., Porter, B., Chaudhri, V., Rodriguez, A., Thomere, J., Mishra, S., Gil, Y., Hayes, P., and Reichherzer, T. (2001). Knowledge entry as the graphical assembly of components. In *Proceedings of the international conference on Knowledge capture*, pages 22–29. ACM Press.

[13] Coffey, J., Reichherzer, T., Owsnicki-Klewe, B., and Wilde, N. (2012). Automated concept map generation from services-oriented architecture artifacts. In Cañas, A. J., Novak, J. D., and Vanhear, J., editors, *Concept Maps: Theory, Methodology, Technology. Proceedings of the Fifth International Conference on Concept Mapping*, Valleta, Malta. University of Malta.

[14] Coffey, J. W. (1999). Institutional memory preservation at NASA Glenn Research Center. Unpublished technical report, NASA Glenn Research Center, Cleveland, OH, USA.

[15] Cunningham, P. (2009). A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Trans. on Knowl. and Data Eng.*, 21(11):1532–1543.

[16] Dai, H. (1997). Discovery of cases for case-based reasoning in engineering. In *Proceedings of the Fourth Asia-Pacific Software Engineering and International Computer Science Conference (APSEC'97)*. IEEE Press.

[17] Davies, J., Duke, A., and Sure, Y. (2003). OntoShare: a knowledge management environment for virtual communities of practice. In *Proceedings of the international conference on Knowledge capture*, pages 20–27. ACM Press.

[18] Denny, M. (2004). Ontology tools survey, revisited. `http://www.xml.com/pub/a/2004/07/14/onto.html`. Accessed Sept. 29, 2013.

[19] Gil, R. and Martin-Bautista, M. (2012). A novel integrated knowledge support system based on ontology learning: Model specification and a case study. *Knowledge-Based Systems*, 36(0):340 – 352.

[20] Hoffman, R. R., Coffey, J. W., Ford, K. M., , and Carnot, M. J. (2001). Storm-LK: A human-centered knowledge model for weather forecasting. In *Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society*, Minneapolis, MN, USA.

[21] Hoffman, R. R., Ford, K. M., Hayes, P. J., and Bradshaw, J. M. (2003). The Borg hypothesis. *IEEE Intelligent Systems*, 18(5):73–75.

[22] Hsieh, S.-H., Lin, H.-T., Chi, N.-W., Chou, K.-W., and Lin, K.-Y. (2011). Enabling the development of base domain ontology through extraction of knowledge from engineering domain handbooks. *Advanced Engineering Informatics*, 25:288–296.

[23] Kendall-Morwick, J. and Leake, D. (2011). A toolkit for representation and retrieval of structured cases. In *Proceedings of the ICCBR-11 Workshop on Process-Oriented Case-Based Reasoning*.

[24] Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

[25] Leake, D., Maguitman, A., and Cañas, A. (2002). Assessing conceptual similarity to support concept mapping. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 168–172, Menlo Park. AAAI Press.

[26] Leake, D., Maguitman, A., and Reichherzer, T. (2003a). Topic extraction and extension to support concept mapping. In *Proceedings of FLAIRS-2002*, pages 325–329. AAAI Press.

[27] Leake, D., Maguitman, A., and Reichherzer, T. (2004a). Understanding knowledge models: Modeling assessment of concept importance in concept maps. In *Proceedings of the Twenty-Sixth Annual Conference of the Cognitive Science Society*, pages 785–790, Mahwah, NJ. Lawrence Erlbaum.

[28] Leake, D., Maguitman, A., and Reichherzer, T. (2005). Exploiting rich context: An incremental approach to context-based web search. In *Modeling and Using Context: 5th International and Interdisciplinary Conference, CONTEXT 2005*, pages 254–267, Berlin. Springer Verlag.

[29] Leake, D., Maguitman, A., Reichherzer, T., Cañas, A., Carvalho, M., Arguedas, M., Brenes, S., and Eskridge, T. (2003b). Aiding knowledge capture by searching for extensions of knowledge models. In *Proceedings of the Second International Conference on Knowledge Capture (K-CAP)*, pages 44–53, New York. ACM Press.

[30] Leake, D., Maguitman, A., Reichherzer, T., Cañas, A., Carvalho, M., Arguedas, M., and Eskridge, T. (2004b). "Googling" from a concept map: Towards automatic concept-map-based query formation. In Cañas, A. J., Novak, J. D., and González, F., editors, *Concept Maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*. Universidad Pública de Navarra.

[31] Lorenzetti, C. M. and Maguitman, A. G. (2009). A semi-supervised incremental algorithm to automatically formulate topical queries. *Information Sciences*, 179(12):1881–1892. Including Special Issue on Web Search.

[32] Maguitman, A., Leake, D., and Reichherzer, T. (2005). Suggesting novel but related topics: Towards context-based support for knowledge model extension. In *Proceedings of the 2005 International Conference on Intelligent User Interfaces*, pages 207–214.

[33] Maguitman, A., Leake, D., Reichherzer, T., and Menczer, F. (2004). Dynamic extraction of topic descriptors and discriminators: Towards automatic context-based topic search. In *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM)*, pages 463–472, New York. ACM Press.

[34] Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M., Cox, M., Forbus, K., Keane, M., Aamodt, A., and Watson, I. (2005). Retrieval, reuse, revision, and retention in CBR. *Knowledge Engineering Review*, 20(3).

[35] Milicic, V., Rizzo, G., Redondo Garcia, J. L., Troncy, R., and Steiner, T. (2013). Live topic generation from event streams. In *Proceedings of the 22nd international conference on World Wide Web companion*, WWW '13 Companion, pages 285–288, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[36] Moon, B. M., Hoffman, R. R., Novak, J. D., and Caas, A. J. (2011). *Applied Concept Mapping: Capturing, Analyzing and Organizing Knowledge*. CRC Press: New York.

[37] Mostek, T., Forbus, K., and Meverden, C. (2000). Dynamic case creation and expansion for analogical reasoning. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 323–329. AAAI Press / The MIT Press.

[38] Novak, J. and Gowin, D. B. (1984). *Learning How to Learn*. Cambridge University Press.

[39] Noy, N., Fergerson, R., and Musen, M. (2000). The knowledge model of Protégé-2000: Combining interoperability and flexibility. In *Proceedings of EKAW 2000*.

[40] Pan, R., Yang, Q., and Pan, S. J. (2007). Mining competent case bases for case-based reasoning. *Artificial Intelligence*, 171(16-17):1039–1068.

[41] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

[42] Santoso, H. A., Haw, S.-C., and Abdul-Mehdi, Z. (2011). Ontology extraction from relational database: Concept hierarchy as background knowledge. *Knowledge-Based Systems*, 24(3):457 – 464.

[43] Shafto, M. and Hoffman, R. (2002). Guest editors' introduction: Human-centered computing at NASA. *IEEE Intelligent Systems*, 17(5):10–13.

[44] Smyth, B. and Keane, M. (1998). Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102(2):249–293.

[45] Smyth, B. and McClave, P. (2001). Similarity vs. diversity. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01*, pages 347–361, Berlin. Springer-Verlag.

[46] Wang, L. and Liu, X. (2008). A new model of evaluating concept similarity. *Knowledge-Based Systems*, 21(8):842 – 846.

[47] Weber, R., Ashley, K., and Brūninghaus, S. (2005). Textual case-based reasoning. *The Knowledge Engineering Review*, 20:255–260.