

Concept Maps and AI: an Unlikely Marriage?

Alberto J. Cañas & Marco Carvalho
Institute for Human & Machine Cognition
Pensacola, FL 32502
www.ihmc.us

Abstract. Concept maps are a graphical representation of a person's (or group of persons') understanding of a domain. As such, it can be considered a knowledge representation scheme. However, the Artificial Intelligence (AI) community frowns on the use of the term "knowledge representation" to refer to concept maps, because they cannot be readily translated to a formal representation for inference or other AI techniques. In this paper we propose that despite the free-style format that concept maps can take, specific characteristics of well-constructed concept maps (structure, semantics, context, etc.) provide an abundance of information on which to develop smart tools that aid the user in the process of constructing concept maps. Our claim is that the compromise in the formalism in lieu of flexibility proposed by concept maps can be compensated, with the help of AI and smart tools, to help bring the best of both worlds to knowledge elicitation and representation. We demonstrate this argument with a set of smart tools that have been implemented in the CmapTools software kit.

Introduction

Concept maps (Cmaps) were developed in the 1970's by Joe Novak (1984) and his research team at Cornell University as a means to help determine how students advanced in their understanding of Science. They are a two-dimensional graphical representation of a person's (or group of persons') understanding of a domain. Cmaps consist of a set of concepts, defined by Novak (*ibid*) as "perceived regularities in events or objects, or records of events or objects, designated by labels", which is constructed so that the interrelationships among them are evident. Concepts are usually enclosed in circles or boxes, and relationships between concepts are indicated by connecting lines that link them together. Words on the linking line specify the relationship between the concepts. The label for most concepts is a single word, although sometimes symbols such as + or % are used. Concept-link-concept triples form propositions, which are meaningful statements about some object or event. Propositions are meant to be semantic units, or units of meaning. In a well constructed Cmap, propositions should "make sense" when read independently. Figure 1 presents a concept map that describes concept maps. A characteristic of Cmaps exemplified in Figure 1 is that the concepts are represented in a hierarchical fashion with the most inclusive, most general concepts at the top of the map and the more specific, less general concepts arranged below. Thus, the vertical axis expresses a hierarchical framework for the concepts. The maps emphasize the most general concepts by linking them to supporting ideas with propositions. The concept "Concept Maps" at the top defines the domain of knowledge pertaining to the map. The triplet "Concept Maps → represent → Organized Knowledge" is a sample proposition. Unless otherwise annotated by arrows, propositions are read from top to bottom.

The Lack of Formalism of Concept Maps

The structure of a Cmap is dependent on its context. Consequently, maps having similar concepts can vary from one context to another and are highly idiosyncratic. The strength of Cmaps lies in their ability to measure a particular person's knowledge about a given topic in a specific context. Thereby, Cmaps constructed by different persons on the same topic are necessarily different, as each represents its creator's personal knowledge. Similarly, we cannot refer to *the* correct Cmap about topic, as there can be many different representations of the topic that are correct.

In educational settings, concept-mapping techniques have aided people of every age to examine many fields of knowledge. When concepts and linking words are carefully chosen, these maps are powerful tools for observing nuances of meaning. Their rich expressive power derives from

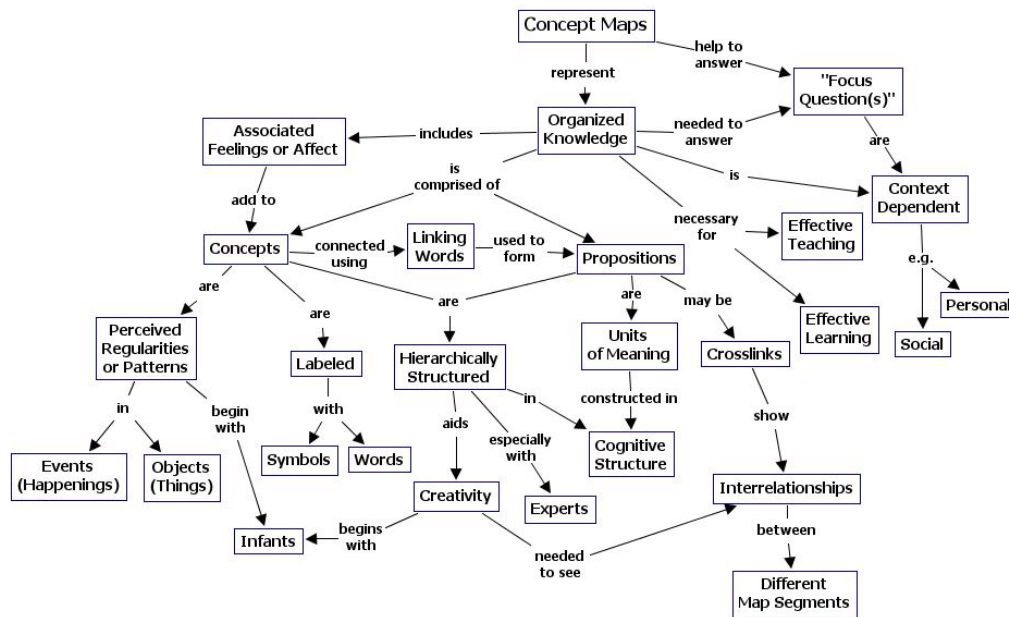


Figure 1. Concept map that describes concept maps.

each map's ability to allow its creator the use of a *virtually unlimited* set of linking words to show how meanings have been developed. However, it is this freedom in the construction of linking phrases in particular that prevents Cmaps from being automatically translatable to any formal representation.

Balance: Usability & Friendliness vs. Formalism

Concept Mapping has had widespread use as a knowledge elicitation (KE) tool. In terms of its yield of propositions that are informative about a domain, concept mapping is at least as efficient as other available KE methods, and is likely the most efficient method for generating models of domain knowledge (Hoffman, Coffey, Carnot, & Novak, 2002). As a KE procedure, it has been successfully employed to form mediating representations and interfaces for intelligent software (i.e., knowledge-based systems and tutoring systems) in a variety of domains (Coffey et al., 2003; Ford, Coffey, Cañas, Andrews, & Turner, 1996). Concept mapping tools have also been successfully incorporated into knowledge elicitation software environments (Ford et al., 1991).

However, Cmaps in their pure form are not suitable for formal knowledge representation because of the freedom provided in selecting concepts – which often are comprised of more than one word – and linking phrases. In the examples cited above, the Cmaps were used as interface to the system, but were not used to directly generate propositions, rules, or any other type of formal representation. Whenever the AI community has taken to using Cmaps for representing knowledge, it is always under the condition that concepts and linking phrases be restricted to predefined taxonomies or ontologies, as in the case of the Shaken environment for large scale capture of expert knowledge (Clark et al., 2001). The resulting formal representations in most cases would probably not be considered Cmaps (in the “Novakian” style) by the concept mapping community.

As a result, most of the AI community frowns on the use of Cmaps and considers them “not interesting” as a means to represent knowledge, or interesting after “formalizing” them. In the AI community, the traditional product of knowledge elicitation tasks is the formal models expected by most AI tools or analysts. That is, knowledge is “translated” from the expert mind into a rigorous and unambiguous representation for further processing. This process, although effective in terms of the end product, is known to cause the “knowledge acquisition bottleneck” (Hayes-Roth, Waterman, & Lenat, 1983). The required formalism often makes it impossible for the expert to construct his/her own knowledge models, requiring the participation of highly training knowledge engineers and a number of interviews for elicitation and model checking.

Concept mapping does not provide the knowledge representation formalism often expected by the AI community as the end product of knowledge elicitation or modeling. It provides a much less rigorous, and yet structured, model of the expert's understanding of the domain. The compromise in formalism reduces the burden on the elicitation process and greatly facilitates the use of the technique by the experts themselves without the intervention of a knowledge engineer (e.g. the Mars 2001 project at NASA Ames by Briggs (2004)), alleviating the knowledge acquisition bottleneck. Concept mapping is thus, a human activity – a means of modeling knowledge by humans in a form that is easily understood by other humans, not by machines.

In our research effort, we are not interested in restricting Cmaps to predefined taxonomies or ontologies in order to obtain a more formal notation, since we believe that the freedom in the selection of concept and linking phrases gives the tool a lot of its power and makes it user-friendly and easy to learn. We argue, instead, that Cmaps offer a reasonable compromise between flexibility and formalism in knowledge representation, and in this paper propose that the structure and context provided by Cmaps enable the development of what we define as *smart software*, described later in this paper.

We have found that following Novak's (1984) guidelines in terms of making concepts single words and linking phrases as short as possible leads to Cmaps that, although not readily translatable to a more formal notation, provide an abundance of information that can be taken advantage of by AI-based *smart* tools that aid the user in the process of constructing Cmaps: instead of trying to convert Cmaps into something they are not in order to use them in AI applications, we have taken AI tools to aid the user in the process of Cmap construction. These tools take advantage of characteristics of Cmaps that we describe in the following section.

Characteristics of Concept Maps

Concept maps have particular characteristics that make them amenable to smart tools. These include:

1. Cmaps have structure: By definition, more general concepts are presented at the top with more specific concepts at the bottom. Other structural information, e.g. the number of ingoing and outgoing links of a concept, may provide additional information regarding a concept's role in the map. (Leake, Maguitman & Reichherzer (2004) present experimental support for the cognitive importance of such factors.)
2. Cmaps are based on propositions: every two concepts with their linking phrase forms a "unit of meaning". This propositional structure distinguishes Cmaps from other tools such as Mind Mapping and The Brain, and provides semantics to the relationships between concepts.
3. Cmaps have a context: A Cmap is a representation of a person's understanding of a particular domain of knowledge. As such, all concepts and linking phrases are to be interpreted within that context.

In addition, in well constructed Cmaps:

4. Concepts and linking phrases are as short as possible, possibly single words.
5. Every two concepts joined by a linking phrase form a standalone proposition. That is, the proposition can be read independently of the map and still "make sense".
6. The structure is hierarchical and the root node of the map is a good representative of the topic of the map.

Smart Tools and CmapTools

At the Institute for Human & Machine Cognition (IHMC) we have developed CmapTools (Cañas et al., 2000; Cañas, Hill et al., 2004) a widely-used software program that supports the construction of Cmaps, as well as the annotation of the maps with additional material such as images, diagrams, video clips and other such resources. It provides the capability to store and

access Cmaps on multiple servers to support knowledge sharing across geographically-distant sites.

We have implemented a set of smart tools into the CmapTools software. These range from a collaboration environment at the “knowledge level” to concept-map based search of the Web. We describe some of these tools as examples of how a smart tool can take advantage of the Cmap characteristics described above.

Soups and the Giant

The Soups (Cañas, Ford, Brennan, Reichherzer, & Hayes, 1995, July; Cañas et al., 2001) provide for a unique type of collaboration among a group of users – usually students – constructing each a map on the same topic. As was stated above, a Cmap can be regarded as an organized collection of propositions relating together a collection of topics. Each proposition is expressed by a simplified sentence which can be extracted from the map by following the arcs’ beginning and end nodes, that is, taking the pairs of concepts and their linking phrases. For example, the map in Figure 2 contains the propositions “Ozone forms an Ozone Layer” and “Ozone Layer is in the Stratosphere”. As the student (for the purposes of this illustration we will assume the users collaborating are students) constructs the Cmap, the system automatically decomposes this map into propositions that are listed in the top window of the pane to the right of the map as shown in Figure 2. This allows two very different (although “logically” equivalent) representations of a student’s ideas: one embedded in the Cmap’s graphical structure and the other more textual in nature.

A student may *publish* a proposition, which makes it potentially visible to other students (published propositions are shown with a pin to the left, as if posted on a bulletin board). We call this process *making a claim*. These published propositions – claims – become part of the collaborating students’ shared “knowledge soup”, which consists of a highly organized “database” of simple claims representing the growing knowledge of the group. It is through these knowledge soups that collaboration and sharing take place. Knowledge soups have many interpretations and can be displayed in several ways. They can be thought of as a body of text, an

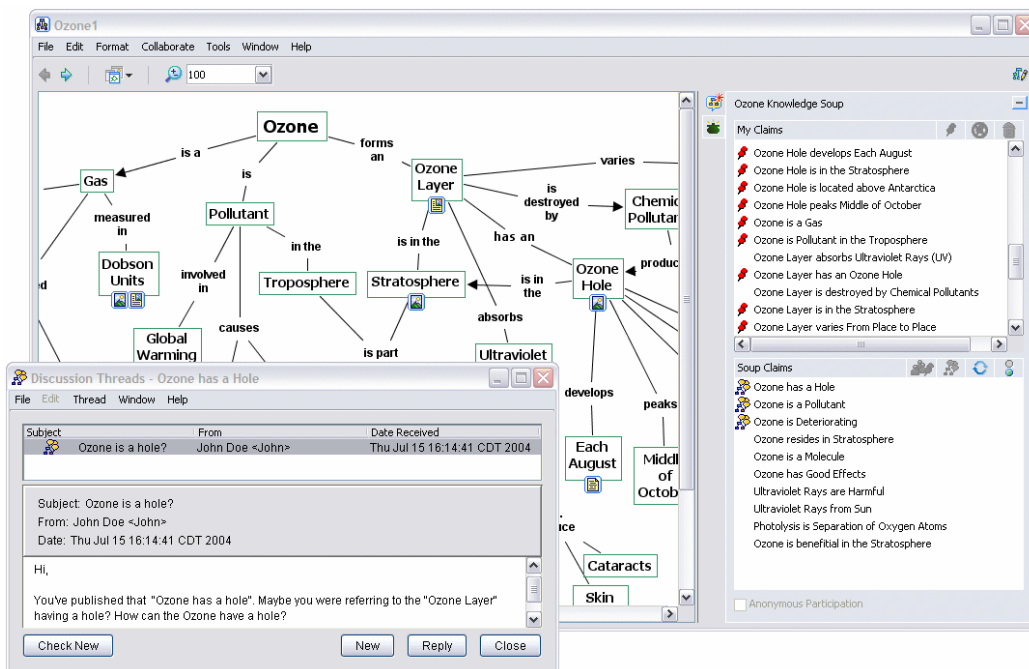


Figure 2: A concept map that is part of a Knowledge Soup on Ozone.

encoding of a larger group Cmap, or an annotated collection of discussions between students.

Published claims can be seen by other students and can be utilized in their own map-building process, but a student can't see *all* claims published by other students, as this would often be cognitively unmanageable. The system has heuristics about the relatedness of knowledge claims. The only claims that a student sees are those directly related to the ones that he contributed to the Soup. As a student publishes more, a wider range of other related claims becomes visible. This strategy is intended to encourage and reward students for participation.

A student can query or question a claim submitted by another student, if he/she disagrees with it or finds it puzzling, and the originator of the claim can respond. Querying a claim causes it to be displayed with a "discussion thread" icon to its left, to indicate to the author and any third parties that it is under dispute or discussion. The querier types a message which becomes visibly attached to the mark. Anyone – including, of course, the originator of the claim – can read this message and respond to it with a further comment, or an explanation or defense of the original claim. In this way, a published claim may become the locus of an extended discussion on some topic. The student's own claims are likewise subject to peer review. In Figure 2, a student has questioned the claim "Ozone has a Hole" – the author of the claim is now expected to respond through this same mechanism.

The Soups are not a communication channel. In fact, the Soup can be set up such that claims are anonymous. The content, not the source, determines which parts of the Soup are displayed to each student. In this way, it is a new kind of group interaction amounting to a collaboration between students entirely in terms of what they know, a collaboration at the knowledge level. Since Soups are accessible across the Internet, the collaboration can take place within a classroom, between classrooms, or between schools, whether local or distant.

The Soups demonstrate the utility of the propositional nature of Cmaps. Even though the propositions may not be translated to a more formal notation, they become the basis of a sharing and collaboration environment.

The Giant

We have developed a software agent – the Giant (Reichherzer, Cañas, Ford, & Hayes, 1998) – that takes advantage of the Soup environment and the propositional nature of the student's beliefs. The Giant, embedded into the program and running independently on each student's machine, generates tentative conclusions derived from both local (the student's) and shared (from other students) propositions from the Knowledge Soup by using a simple set of rules. It presents its own claims to the student and asks for decision upon their soundness.

We referred to the agent as a Giant because in a sense it 'knows' a great deal but it completely lacks common sense knowledge and has limited reasoning capabilities that sometimes induce a silly but amusing behavior. The Giant is not guaranteed to draw rational conclusions from the Cmap (as a human being would) and in no way does it verify the student's propositions – such an intention is not pursued in our study. However, the Giant's propositions often act as a 'destabilizer' to the student, proposing conclusions that encourage the user to consider a different line of thought.

The Giant is based on small set of rules which can be classified into three categories: transitivity; quantifier, qualifier and dependencies; and classification and extension. Based on these rules and the local and soup propositions, the Giant generates its own conclusions. For example, if the student has a proposition "plants have leaves" and another student has "leaves are green" in the Soup, the Giant concludes that "plants are greens" and proposes this as a claim to the student (the rules check not only for matching concepts, but also for pairs of linking words that indicate reasonable conclusions). If the student has "Some stars are neutron stars", the Giant notes the quantifier "some" and will conclude that "There are stars that are not neutron stars", or "The ability to be a neutron requires something", or "Stars are not always neutron stars". Figure 3

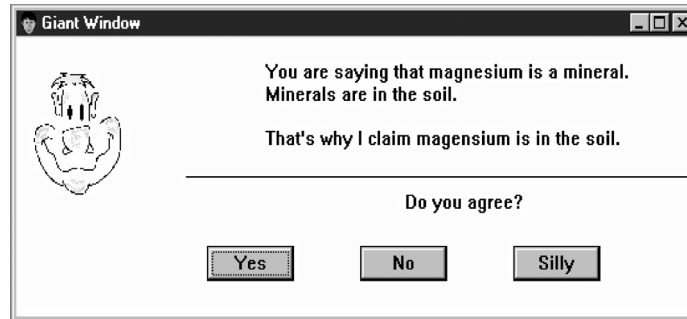


Figure 3: The Giant proposes a claim to the student.

shows the Giant taking the student’s claim “Magnesium is a mineral” and a claim from the Soup “Minerals are in the soil” to conclude that “Magnesium is in the soil”. The Giant’s claims are displayed in a separate window, and when the student clicks on one of them, a dialogue box like that in Figure 3 is displayed, whereby the student can teach the Giant whether the claim is true, false, or silly. The silly option shows that the Giant, even though it has access to all the claims in the Soup, is likely to come up with irrational conclusions. But since the Giant is not meant to teach the student, it doesn’t really matter if he is not always right.

The Giant further demonstrates how smart tools can take advantage of the propositional nature of the Cmaps. With limited capabilities of sentence understanding and construction, the Giant is able to help the students construct better Cmaps.

Word Disambiguation – What is this Concept Map About?

Words often have more than one meaning, and a difficult task in natural language processing is determining the correct meaning of a word in a text based on its context. Because of its free-form nature, this ambiguity is also present in Cmaps, as can be seen in the two simple examples in Figure 4 that show two maps using the concept “chair” with two different meanings. Being able to determine which the correct meaning of “chair” is in each of the Cmaps would help other tools, e.g. the Giant, make better decisions.

In Cañas et al. (2003) we have reported on an effort to use WordNet (Miller, 1990) to disambiguate the sense of words in Cmaps, whether they are part of a concept or a linking phrase. By exploiting the topology and semantics of the Cmaps, the algorithm uses the senses and semantic relations provided by WordNet to try to determine which of the senses in WordNet best matches the context of the Cmap. To disambiguate a word w , the algorithm begins by selecting key concepts from the map: (a) words in concepts that are in the same proposition w , (b) words in the root concept of the map, and (c) other words in the concept to which w belongs.

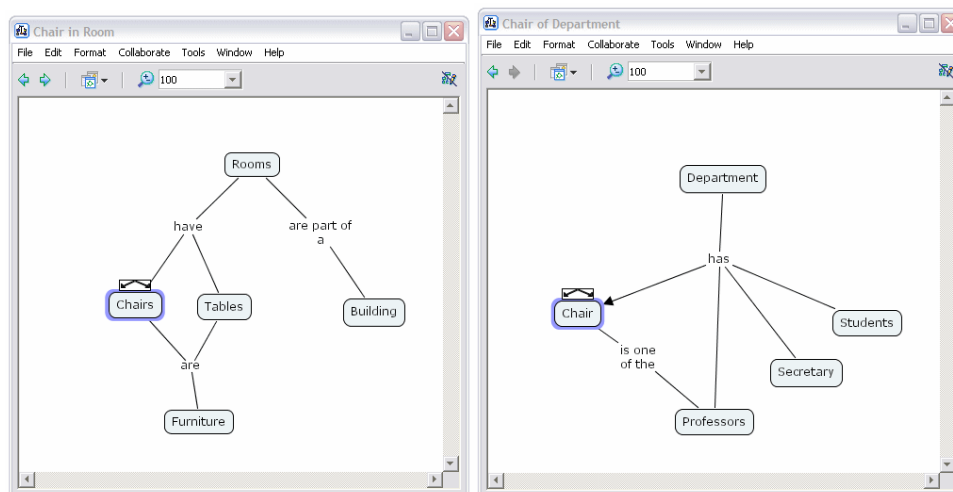


Figure 4. Two concept maps using the same word “chair” with different meanings.

These selected words and w are then used in a series of steps described in Cañas et al. (2003): (a) each of the words are related to one or more WordNet synsets (a synset is a set of synonym words representing a concept in WordNet), (b) all possible hypernym sequences are constructed whose last element is a synset in one of those sets, (c) clusters of hypernym sequences are created, and (d) the best cluster is selected, and (e) the last synset of the selected cluster provides the disambiguated sense of the word. Other algorithms used for disambiguating words in text (e.g. Fellbaum et al. (2001) have the problem of selecting the key words, which in the case of text is difficult because there is no particular structure, and the relation between the words is not clear. Our algorithm exploits the topology of the map by including only the words of key concepts as part of the disambiguation process.

Figure 5 shows the result of disambiguating the concept “chair” in the Cmaps in Figure 4. For each of the map, the “Chair” concept was selected, and the WordNet server available for CmapTools was invoked to display information on the term. The invocation of this function automatically executes the word disambiguation algorithm described above and, as a result, the correct sense of the word is shown as the top entry in the Definitions list. The top window shows the result for the “Chair in Room” map and the bottom window for the “Chair of Department” map. As can be seen, the algorithm selected the correct meaning in both cases.

In Cañas et al. (2003) we reported on an experiment where subjects were asked to select the

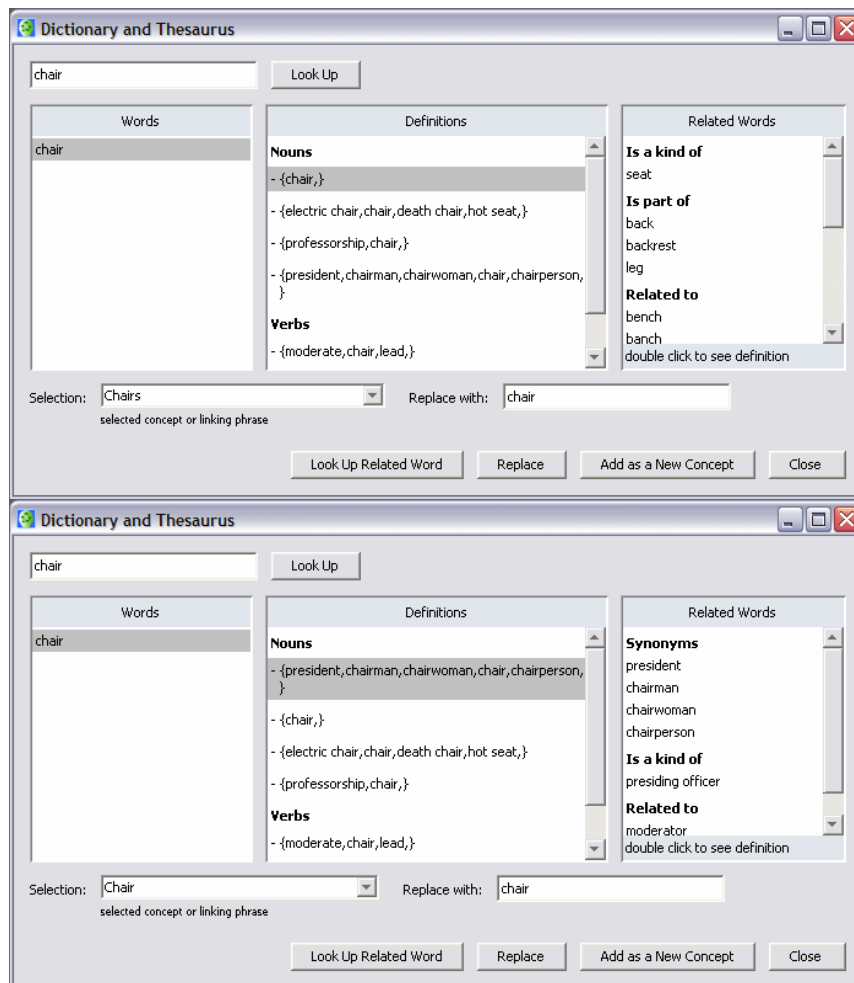


Figure 5: Disambiguating the sense of “chair” in the Cmaps in Figure 4.

appropriate sense for concepts in Cmaps. The results showed that the algorithm was able to agree with the sense selected by the subjects in 75% of the cases.

Being able to disambiguate the sense of a concept opens a whole new set of possible smart tools to apply to Cmaps. Two of these tools, which do not yet take advantage of this algorithm, are presented in the next section.

Searching the Web based on a Concept Map

Search engines have become a hot topic as the Web grows into what a few years ago was an unimaginable size. The act of web browsing has been replaced lately with large scale search engines. Nobody “surfs” the web any more: users Google into the topic they are interested and then leave. That is, many Web users rely on the large scale server farm indexing billions of Web pages for them.

However, search engines rely on the query specified by the user, and users rarely provide a “good” query: the average search query, from a recent study, is 2.2 words (Spink, Wolfram, Jansen, & Saracevic, 2001). As a result, the list of retrieved documents is often too large or contains information that has no relevance to the query. We propose to alleviate this problem by taking advantage of the context provided by Cmaps to both (a) provide more complete queries to the search engines, and (b) enhance the ranking of the results provided by the engines.

The user can easily and concisely specify the context of the search in a Cmap, which will be used for the automatic construction of queries. The web-search algorithm implemented in CmapTools allows the user to select a concept and ask the system to search for Web information that is relevant to the concept within the context of the Cmap. The process consists of: (a) Analyzing the Cmap to prepare a relevant query to use in searching the Web, (b) Retrieving relevant documents from the Web, (c) Ranking the retrieved web pages according to relevance, and (d) Presenting the results to the user. We’ll briefly describe each of these steps. A more detailed explanation can be found in Carvalho et al. (2001). To generate the query, key concepts are selected from the map. These include the words in the selected concept itself, the root of the Cmap, and authority nodes: those with the highest number of outgoing links to other nodes. We assume that the number of outgoing links is an indicative of further elaboration of these concepts, and therefore a gauge of their relevance in the context of the map. We use the query constructed from the key concepts in the previous step to retrieve Web pages and build our collection of documents. We have developed a meta-search engine, based primarily on Google (Brin & Lawrence, 1998), in order to retrieve an initial set of documents from the public Internet. Once retrieved, these documents are added to a local cache for ranking, which is based on comparing distance matrices calculated from the Cmap and from each of the candidate documents. The distance matrices are symmetric, and provide a weighted list of the concept terms in the map and the documents. Weights between terms in the Cmap are proportional to the number of linking phrases between each concept. For the documents, the weights are estimated as a function to the number of words between each Cmap term found in the text. Lower distances between terms represent a higher weight, as terms are more likely to form a proposition.

Experimental results reported in Carvalho et al. (2001) show that the algorithm scored similarly or better than the best of four publicly available search engines in ranking of retrieved documents for relevance to the Cmap according to subjects’ criteria, and clearly performed better than the other three. The combination of leveraging on the structure of the Cmap in generating the query, and utilizing the propositional nature and hierarchical topology of Cmaps to provide contextual information to identify and rank retrieved documents that are more relevant seems to provide an improvement over the ranking provided by publicly available search engines.

Smart Tools for Building Concept Maps

CmapTools web search capabilities can also be used to aid the user in the process of building a Cmap. The CmapTools application proactively monitors the context of an open Cmap to

autonomously search for web information that could be relevant to the user. At the user's discretion, this information can be utilized to verify, correct or extend his/her Cmaps during browsing or authoring.

This capability is integrated with the editor, and Web-based suggestions can be provided in two forms: a) a list of relevant Web-pages that can be used as references by the user or b) a list of relevant concepts that the user can, at his/her discretion, add to the Cmap to broaden or clarify the content. This information is proactively offered by the application as the map context changes, but it can also be requested on-demand by the user.

The suggestion of Web-pages is based on an enhanced version of the search algorithm proposed to obtain relevant pages from the Web based on the current stage of the Cmap (Leake et al., 2004). A key difference in the proactive suggestion of Web-pages is that the whole map is taken into account. This is in contrast to the on-demand search, where the user is allowed to specify a concept within the map that will be used to focus the search.

CmapTools can also suggest concepts during the construction of a map. Cmap construction is a meaning-making process in which listing the concepts that will be included in the map is a less central task than selecting the appropriate linking phrase to form propositions. Often, however, we have found that users struggle to "remember" new concepts to add to their maps, and we believe that they should be able to concentrate their efforts of determining the links between concepts in the map. We have implemented in the CmapTools software a proactive *concept suggester* (Cañas, Carvalho et al., 2004) module that, during map construction, analyzes the Cmap, creates a relevant query to search the Web for documents related to the map, extracts relevant concepts from the retrieved Web pages, and presents the results as suggestions to the user. This module searches for new suggestions whenever it determines that the map has suffered significant changes.

Cañas et al. (*Ibid*) described tests of this module with a group of users, and have reported that the results indicate that the module is effective in presenting relevant concepts to the users. This effectiveness, however, diminishes as the map grows, which implies that the algorithm should be revised to take into account that in larger maps, users are most likely working on a piece of the map, and so suggested concepts should be determined by the context of that piece. In this experiment, subjects (13 total) were given a topic for a Cmap to be built with a minimum of 20 concepts. During the construction of the map, the subjects were periodically prompted with a list of suggestions for concepts. The suggestions were automatically mined from the web by the CmapTools application based on context extract from different stages of the map. When prompted, each user was asked to rank the concepts in a scale of relevance. On average each subject was prompted 6 times with a list of 15 suggestions (each time) during the construction of a map. Up the third stage ranking the percentage of subjects that ranked at least 4 of the 15 concepts suggested as relevant was above 62% and the number of subjects that ranked at least 3 of them as relevant was above 77%. For later stages, as the map grew in size, the number percentages drop but stayed always above 50%.

The CmapTools modules to mine the web in search of relevant Web pages and concepts to aid the user during Cmap construction are exemplars on how smart tools can leverage on the particular characteristics of Cmaps.

Conclusions

Cmaps are two-dimensional graphical representations of a person's understanding of a domain. Recognized (and widely used) as efficient tools for knowledge elicitation, representation and sharing, Cmaps are often criticized by the AI community for the lack of formalism that is intrinsic to the technique. In this paper we have shown that, even without compromising the flexibility proposed in concept mapping, it is possible to design smart software tools that leverage from the Cmap structure, hierarchy and context to accomplish complex inference tasks.

Our claim is that the compromise in the formalism in lieu of flexibility proposed by Cmaps can be compensated, with the help of AI and smart tools, to help bring the best of both worlds to knowledge elicitation and representation.

Acknowledgements

This work was partially funded by NASA's Intelligent Systems Program under awards NCC 2-1216 and 2-1297 and by the US Navy's Chief of Naval Education and Training.

References

- Briggs, G., Shamma, D., Cañas, A. J., Carff, R., Scargle, J., & Novak, J. D. (2004). Concept Maps Applied to Mars Exploration Public Outreach. In A. J. Cañas, J. D. Novak & F. González (Eds.), *Concept Maps: Theory, Methodology, Technology, Proceedings of the 1st International Conference on Concept Mapping*. Pamplona, Spain: Universidad Pública de Navarra.
- Cañas, A. J., Carvalho, M., Arguedas, M., Leake, D. B., Maguitman, A., & Reichherzer, T. (2004). Mining the Web to Suggest Concepts during Concept Map Construction. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology, Proceedings of the 1st Int. Conference on Concept Mapping*. Pamplona, Spain: Univ. Pública de Navarra. Pamplona.
- Cañas, A. J., Ford, K. M., Brennan, J., Reichherzer, T., & Hayes, P. (1995, July). *Knowledge Construction and Sharing in Quorum*. Paper presented at the Seventh World Conf. on Artificial Intelligence in Education, Washington DC.
- Cañas, A. J., Ford, K. M., Coffey, J. W., Reichherzer, T., Carff, R., Shamma, D., & Breedy, M. (2000). Herramientas para Construir y Compartir Modelos de Conocimiento basados en Mapas Conceptuales. *Revista de Informática Educativa*, 13(2), 145-158.
- Cañas, A. J., Ford, K. M., Novak, J. D., Hayes, P., Reichherzer, T., & Niranjana, S. (2001). Online Concept Maps: Enhancing Collaborative Learning by Using Technology with Concept Maps. *The Science Teacher*, 68(4), 49-51.
- Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gómez, G., Arroyo, M., & Carvajal, R. (2004). CmapTools: A Knowledge Modeling and Sharing Environment. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology, Proceedings of the 1st International Conference on Concept Mapping*. Pamplona, Spain: Universidad Pública de Navarra.
- Cañas, A. J., Valerio, A., Lalinde-Pulido, J., Carvalho, M., & Arguedas, M. (2003). *Using WordNet for Word Sense Disambiguation to Support Concept Map Construction*. Paper presented at the Proceedings of SPIRE 2003: International Symposium on String Processing and Information Retrieval, Manaus, Brasil.
- Carvalho, M. R., Hewett, R., & Cañas, A. J. (2001). *Enhancing Web Searches from Concept Map-based Knowledge Models*. Paper presented at the SCI 2001: Fifth Multi-Conf. on Systems, Cybernetics and Informatics, Orlando.
- Clark, P., Hayes, P., Reichherzer, T., Thompson, J., Barker, K., Porter, B., Chaudhri, V., Rodriguez, A., Thomere, J., Mishara, S., & Gil, Y. (2001). *Knowledge Entry as the Graphical Assembly of Components*. Paper presented at the Int. Conference on Knowledge Capture, Victoria, BC, Canada.
- Coffey, J. W., Cañas, A. J., Reichherzer, T., Hill, G., Suri, N., Carff, R., Mitrovich, T., & Eberle, D. (2003). Knowledge Modeling and the Creation of El-Tech: A Performance Support System for Electronic Technicians. *Expert Systems with Applications*, 25(4).
- Fellbaum, C., Palmer, M., Dang, H., Delfs, L., & Wolf, S. (2001). *Manual & Automatic Semantic Annotation with WordNet*. Paper presented at the Workshop on WordNet and Other Lexical Resources, NAACL-01, Pittsburgh, PA.
- Ford, K. M., Cañas, A. J., Jones, J., Stahl, H., Novak, J. D., & Adams-Webber, J. (1991). ICONKAT: An integrated constructivist knowledge acquisition tool. *Knowledge Acquisition*, 3, 215-236.
- Ford, K. M., Coffey, J. W., Cañas, A. J., Andrews, E. J., & Turner, C. W. (1996). Diagnosis and Explanation by a Nuclear Cardiology Expert System. *Int. Journal of Expert Systems*, 9, 499-506.
- Hayes-Roth, F., Waterman, D., & Lenat, D. (Eds.). (1983). *Building Expert Systems*: Addison-Wesley.
- Hoffman, R. R., Coffey, J. W., Carnot, M. J., & Novak, J. D. (2002). *An Empirical Comparison of Methods for Eliciting and Modeling Expert Knowledge*. Paper presented at the Meeting of the Human Factors and Ergonomics Society, Baltimore MD.
- Leake, D., Maguitman, A., Reichherzer, T., Cañas, A. J., Carvalho, M., Arguedas, M., & Eskridge, T. (2004). Googling from a Concept Map: Towards Automatic Concept-Map-Based Query Formation. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept Maps: Theory, Methodology, Technology, Proceedings of the First International Conference on Concept Mapping*. Pamplona: Universidad Pública de Navarra.
- Leake, D. B., Maguitman, A., & Reichherzer, T. (2004). *Understanding Knowledge Models: Modeling Assessment of Concept Importance in Concept Maps*. Proc. of the Twenty-Sixth Annual Conf. of the Cognitive Science Society.
- Miller, G. (1990). Five Papers on WordNet. *International Journal of Lexicography*, 3(4).
- Novak, J. D., & Gowin, D. B. (1984). *Learning How to Learn*. New York: Cambridge University Press.
- Reichherzer, T. R., Cañas, A. J., Ford, K. M., & Hayes, P. J. (1998). *The Giant: A Classroom Collaborator*. Paper presented at the Proceeding of the Fourth Int. Conference on Intelligent Tutoring Systems (ITS). San Antonio.
- Spink, A., Wolfram, D., Jansen, B., & Saracevic, T. (2001). The Public and Their Queries. *Journal of the American Society for Information Science and Technology*, 52(3), 226-234.