Augmenting Human Intelligence via Externalized Knowledge Representation and Intelligent Information Retrieval

by

Gleb Kuznetsov

B.S., Massachusetts Institute of Technology (2010)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2011

Author ....
Department of Electrical Engineering and Computer Science
Feb 1, 2011

Certified by ....
Patrick H. Winston
Ford Professor of Artificial Intelligence and Computer Science
Thesis Supervisor

Accepted by ....
Dr. Christopher J. Terman
Chairman, Master of Engineering Thesis Committee

Augmenting Human Intelligence via Externalized Knowledge Representation and Intelligent Information Retrieval

by

Gleb Kuznetsov

Submitted to the Department of Electrical Engineering and Computer Science
on Feb 1, 2011, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

With a representation of a user's mental model in hand, a computer system can continuously query against a knowledge base of past work sessions or information on the Web and generate a set of recommended resources for the user to consider. In this thesis, I have developed an interface and a representation that allows a computer system to build a model of a user's intent and generate recommendations. I have designed, prototyped, and deployed the Mental Model Browser, a web application that infers a user's intent during a Web browsing session and provides recommendations for related URLs. The application includes a web browser extension for recording the URLs a user visits and a feedback interface that hosts a dialogue between the computer system and the user. The Mental Model Browser identifies important concepts in the session by leveraging an API provided by the Delicious web bookmarking service, a rich data corpus of crowd-sourced web page tags. The identified concepts are presented to the user to confirm their validity and trigger a query for recommended web pages. I conducted a pilot study with 22 active participants who engaged in 56 web browsing sessions. The results of the study show that users were able to readily adapt to the workflow of the application. Users reported quickly discovering how to help shape the system's model of the session through the use of tags. Several users reported receiving valuable recommendations that they did not find through search alone. Finally, I lay out visions for near-future technologies such as multi-modal knowledge capture and activation, as well as knowledge-based social networks that are enabled by the concepts I have explored in this thesis.

Thesis Supervisor: Patrick H. Winston
Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgments

I will never be able to adequately repay Paul Keel for the role he has played as a mentor to me these past two years. As a designer, engineer, manager, and philosopher, his wisdom and advice have provided the direction for my intellectual exploration. From our initial discussions about philosophy of mind to helping me clarify the most compelling ideas in my thesis, Paul has been an impeccable guide and friend.

The catalyst for my entrance into artificial intelligence research was none other than the charismatic pioneer of the field Patrick Winston. His lessons in identifying what is important in the research out there, and how to package and sell your own ideas have been an invaluable resource in reaching this point, and undoubtedly will continue to guide me in life. Through his patience and insight, he helped me translate my chaotic ideas into plans for action. And his initiative to introduce me to Paul Keel provided a key pivot point in my life.

Marvin Minsky's writings and seminar had an immense influence on me. He taught me how to think about thinking, and his ideas have fused into the core of my own philosophy that guides me in research and life.

The summer I spent as a UROP with Bo Morgan and Dustin Smith helped me discover my creative and productive potential. You guys epitomize the academic optimism that leads to great breakthroughs. The conversations we continue to share have been incredibly meaningful and have helped me develop my confidence to take on tough problems.

To all of my friends who have tolerated my philosophical meandering over the years, and told me to suck it up when I was down: Much appreciated. I feel like I'm able to live dozens of parallel lives through you guys, all of your ways of thinking, failures, and successes have been invaluable lessons to me. If we look at life as assembling parts into a product that is our Selves, you guys have provided a greater selection of parts than I could ever use.

But of course nothing would have been possible without my parents. You taught me to avoid the crowd and to question authority. Whatever compelled you to pick up and leave the Soviet Union back in 1989, a rambunctious 2-year old child in tow, I cannot imagine ever having the incredible opportunities I have had otherwise. My father has provided deep insight and advice with perfect timing throughout my life, and continues to keep me one step ahead of the world's challenge. I could not imagine a better father. If I can one day serve as half the mentor he has been to me, I will feel complete. My mother's ability to keep me thinking positively and pushing me to take advantages of opportunities around me has kept me ever moving forward in my life. I will never know a stronger person. Returning home for winter break of my freshman year and learning of her cancer diagnosis derailed my once impervious immunity to emotional challenges. But seeing her overcome the disease over the next two years and emerge stronger and healthier than before has provided the inspiration and evidence to me that nothing can stop the determined spirit. We only get a short time with those that we love, and the importance of family is thing I will ever come to be sure of besides the undeniable fact that we exist.

Contents

List of Figures

Chapter 1

Introduction

1.1 Vision

The dream of augmenting human intelligence through computational means has had a growing following for over half a century now since the advent of electronic computers. Human and machine each has his own respective strengths: Computers are designed for performing exhausting calculations and algorithmically sifting through large amounts of data. Humans still have the upper hand, however, in identifying structural relationships and patterns in the world around them. Together, man and machine can work together to solve problems of ever-increasing complexity. The design of the interface between man and machine is key to unlocking greater degrees of symbiosis.

Douglas Engelbart captures the spirit of this endeavor in his vision [Engelbart, 1962]:

> By "augmenting human intellect" we mean increasing the capability of a man to approach a complex problem situation, to gain comprehension to suit his particular needs, and to derive solutions to problems. Increased capability in this respect is taken to mean a mixture of the following: more-rapid comprehension, better comprehension, the possibility of gaining a useful degree of comprehension in a situation that previously was

> too complex, speedier solutions, better solutions, and the possibility of finding solutions to problems that before seemed insoluble ... We refer to a way of life in an integrated domain where hunches, cut-and-try, intangibles, and the human "feel for a situation" usefully coexist with powerful concepts, streamlined terminology and notation, sophisticated methods, and high-powered electronic aids.

We have since experienced several paradigm shifts in computation and communication, but the present relationship between man and machine still leaves plenty to be desired. In an age defined by the omnipresence of a World Wide Web, our interactions with computers remain largely linear. We are able to represent and manipulate pieces of information as individual documents, yet we store and retrieve them as lists in electronic folders. Search engines project our queries onto narrow planes of the web, yet we must still laboriously sift through the results to find, often only approximately, what we need. We are able to exchange messages with anyone in the world, yet we must seek out our would-be good friends and collaborators manually - our networks hardly extend beyond the connections we make in person.

Now, we're at a point where we have the technological tools and the theoretical wisdom to realize true "man-machine symbiosis" [Licklider, 1960]. It is a matter of designing the right human computer interfaces and engineering well-calibrated implementations of methods available to us.

> **I believe the next step is to give computers a way of understanding what is going on inside a user's head when he or she is solving a problem.**

I have designed, prototyped, and deployed the Mental Model Browser application in order to explore means of augmenting human intelligence in the familiar domain of searching for information on the web. The present investigation offers to grant insight into techniques for capturing and augmenting problem solving that cater to the way our minds work. And as a result, we may gain insight into intricacies of our intelligence.

1.2 Mental Model Browser

The Mental Model Browser is a web-based software application for augmenting the typical experience of seeking information on the web. I developed the system in order to investigate how to provide a computer with a model of the user's intent and knowledge, and to investigate the effects of the resulting dialogue between the computer and the user on the user's experience.

I narrow my perspective on knowledge-seeking as happening in discretized bursts, or sessions. Each session is characterized by a goal or question, a period of information gathering, and a means of crystallizing the knowledge learned (through an action, a written record, or a mental resolution). I have designed, prototyped, and conducted initial user tests of a system that combines elements of mainstream and experimental tools for establishing a channel of communication between the human and machine during web browsing.

As the user searches the Web, the system constructs a representation of the user's session, interacts with the user to clarify the salient concepts of the session, and provides recommendations based on its growing understanding.

1.2.1 Definitions

- **Session**: a knowledge-seeking period identified by a goal.
- **Resource**: a document, generally identified by its URL, accessed during a Session.
- **Tag**: a label on a Resource or Session that identifies a key idea or concept in that Resource or Session; may also serve as a reminder to the user about the problem being solved when the Tag was created.
- **Topic Model**: the collection of all Tags associated with a particular Resource or Session.

1.2.2 Features

- **Browser Extension (Google Chrome only for now)**: enables passive recording of a browser session by sending data about URLs a user visits to a server.
- **Session View User Interface**: the dashboard for recording browsing sessions. The UI allows for viewing the present session progress, adding tags, and interacting with machine recommendations.
- **Recommendation Engine**: a system for constructing representations of the user's history and actions and providing suggestions composed of potentially related URLs that the user hasn't seen yet.
- **History View User Interface**: allows the user to access previous session data; an externalized record of the knowledge-seeking process.

1.2.3 User Scenario

To illustrate the function of the system, I walk through a typical user session, focusing on the user experience and touching upon the back-end functionality.

1. *(First time only)* The user installs the required extension for the Google Chrome web browser at the experiment landing page: `http://singularity.xvm.mit.edu`.
2. The user comes up with a session goal (e.g. a question to answer, problem to solve, or skill to acquire) (Figure 1-1)
3. The user presses the record button and enters his goal into the interface to label the session (Figre 1-2).
4. The user begins by conducting a typical web search. (Figure 1-3)

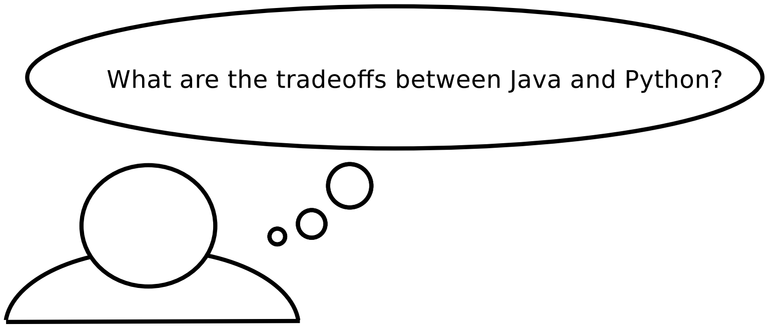


Figure 1-1: User has a question

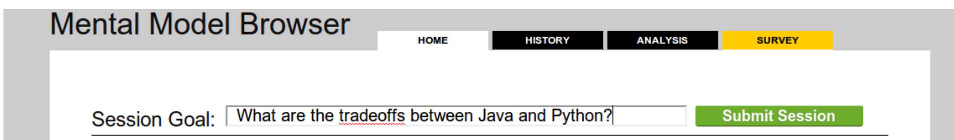


Figure 1-2: User enters goal

Figure 1-3: Perform Search

5. As the user accesses resources, the browser plugin sends data about the resources accessed to the experiment server. (Figure 1-4)

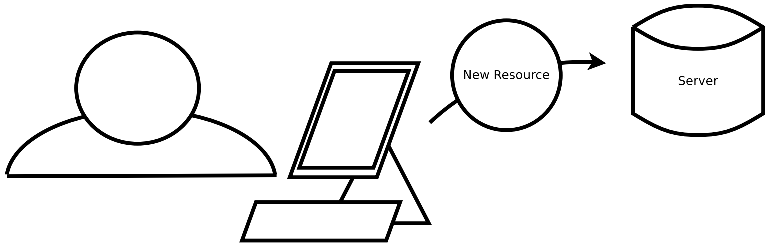


Figure 1-4: Send Resource

6. The server receives information about the user's behavior and begins to make inferences about the important ideas. The system provides feedback in the form of resources visited, recommended tags (the important ideas), and recommended resources. (Figure 1-5)

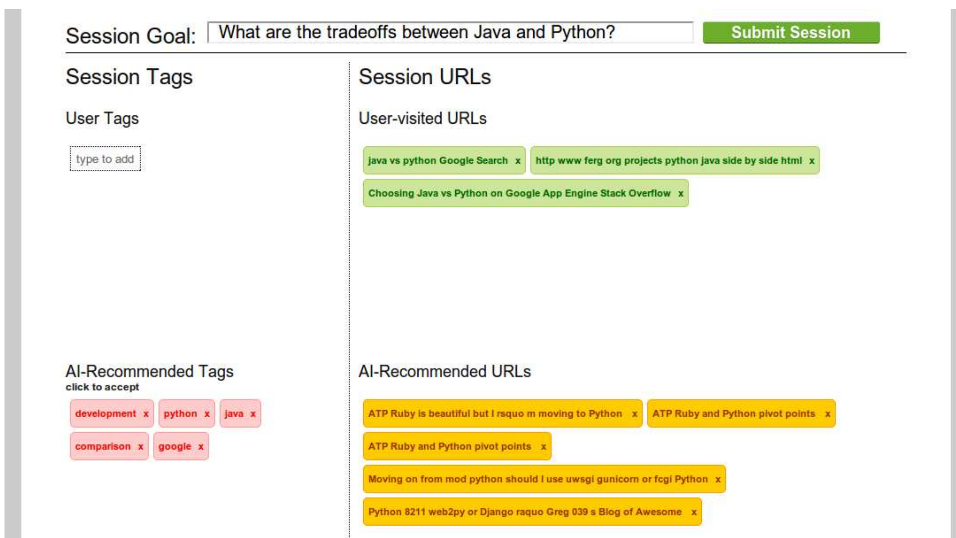


Figure 1-5: Feedback UI

7. The user responds to the system by accepting correct tags and adding his own that the system didn't come up with. (Figure 1-6)

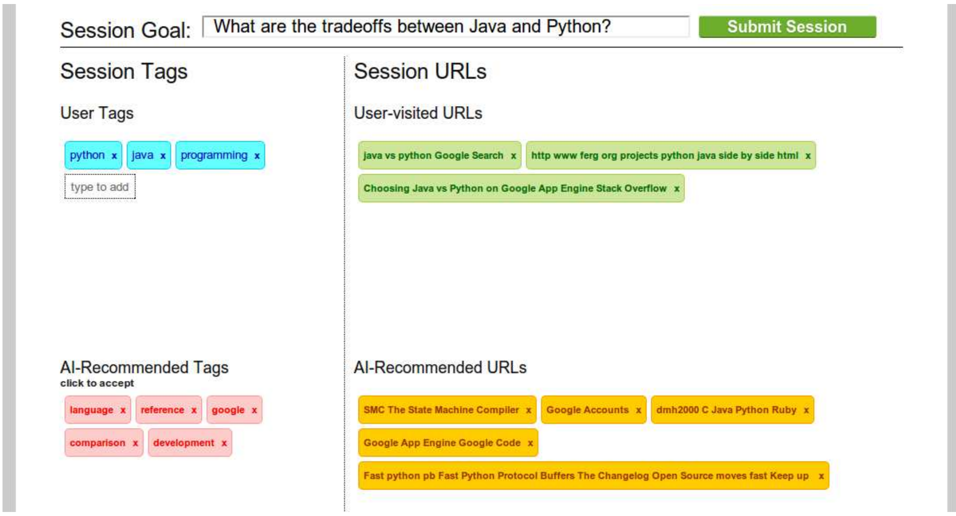


Figure 1-6: User adds Tags

8. The user discovers recommended resources that answer the present goal and suggest related explorations. (Figure 1-7)

Figure 1-7: Recommendations

1.2.4 Comparison to Typical Web Browsing

The Mental Model Browser system augments the typical web browsing experience in several ways:

Allows the user to visualize the entire browsing session.
Users usually do not view their history as they are browsing so can be easily distracted from the goal. By viewing the path taken so far, as well as the model of the session being constructed, the user is able to stay focused on the goal and potentially have a

more productive session.

Provides recommendations from external web sources based on an understanding of the session.

The system infers the user's intent from the observing the web URLs visited and the feedback provided by the user adding tags and provides additional recommendations, both in the form of tags and recommended URLs that the user may not have discovered otherwise. A search engine can only give a response to a query. The Mental Model Browser turns the session thus far into a query in itself.

Engages in dialogue between the user to assess what is important and help the user express his intent.

The system provides recommendations in the form of tags that not only aid in keeping the user focused, but provide a way for the user to convey what are the important parts of the session. This improves the quality of the recommendations and provides better hooks for future activation of this session and its resources.

Forms a history based around a specific goal and collection of web documents related to that goal.

The present web history is simply a list of resources visit according to time. Even if a user searches his web history the lack of context makes it difficult to find what is important. The Mental Model Browser, on the other hand, organizes the history into discrete goal-oriented sessions. Additionally resources can be activated according to the important concepts in their respective sessions.

1.2.5 Enhancing Delicious

The Delicious bookmarking service as a product provides some inspiration for the ideas in this thesis. I've used the service regularly over the past few years in order to remember interesting web pages that may serve as resources in the future. However, Delicious does not allow for capturing entire sessions as readily, so the connection between the problem being solved at the time and the web document is often lost.

The system that I have developed, the Mental Model Browser, may be viewed as an enhancement to the Delicious service that not only adds functionality but moves the usage pattern closer to the way humans think and organize knowledge. Instead of bookmarking and tagging individual pages, my system allows for bookmarking collections of resources defined by a common goal, and applying tags to the session as a whole. In fact my system could be reduced to serve the same functionality as Delicious, if each session were reduced to observing a single resource. Of course, the benefits of my system are based on sessions composed of several resources so the trivial equivalence to Delicious does not receive much attention.

Another benefit to the Mental Model Browser is the prevalence of suggestion. In Delicious there are often recommended tags when bookmarking a page, but they are limited to that document alone. In the system I've developed, the suggestions are arguably more useful because they derive from the session and so point more directly to the problem being solved at the time of bookmarking. These tags then become more valuable hooks for retrieval of the session and resources.

1.3 Externalizing Cognition

The Mental Model Browser system is a first step in designing more ambitious systems for externalizing cognition. A major benchmark toward achieving a greater degree of man-machine symbiosis would be to construct a system that is capable of dynamically capturing a multi-modal sample of users' problem solving tasks and provide means for access and manipulation of the captured data for aid in future tasks. Voices in the cognitive science literature argue that the power of human problem solving resides in the interplay between internalized and externalized knowledge [Kirsch, 2009]. Thus, if the knowledge in a problem solving session can be represented in a machine-understandable form, it is possible to apply computational techniques to aid humans in the manipulation of information, taking us beyond the limited tools available to us now. Electronic capture of problem solving is very feasible today, as these days most research and design processes occur within a computational environment. Further, in many fields, such as computer programming and finance, most of the actual synthesis and decisions are executed within a computational environment as well.

> **The question, then, is how can we most effectively capture these processes, construct useful representations, and augment human problem solving?**

The implications of having a machine-readable representation include:

- Machine-queryable database of human processes and knowledge
- Machine-curated pattern and relationship discovery.
- Machine-curated information retrieval from the World Wide Web or other networked data sources.

1.4 Contributions

Mental Model Browser

Designed and built the Mental Model Browser, a software system for augmenting the typical web browsing experience. The system provides a mechanism for tracking browsing data, constructing a representation, carrying on a dialogue with the user, and providing recommendations based on the system's understanding of the user's intent.

Topic Model Representation

Implemented a topic model representation for abstracting the content of individual web documents, entire browsing sessions, and users. The topic model representation allows for the system to predict structural similarity among data objects and provide recommendations the user.

Applied Research Philosophies to Web Browsing Context

Applied cross-disciplinary ideas spanning across artificial intelligence, cognitive science, and information technology to the context of web searches. Worked at the intersection of the EWall and Genesis projects to experiment with salient ideas for enhancing web browsing.

Pilot Study

Deployed the system to 22 active users who engaged in 56 browsing sessions from which an analysis was carried out to determine how the system was used and whether it provides an experience that more closely matches innate ways of thinking.

Chapter 2

Background

2.1 Related Projects

I was fortunate to find myself at MIT in the wake of the EWall Project and the rise of the Genesis Project. My thesis was conducted at the middle ground of these two projects. The respective project leaders, Dr. Paul Keel and Professor Patrick Winston, were my direct advisors and the resulting application is as much a result of the conversations I've had with them as with constructing prototypes and finding out what does and doesn't work. I describe the backgrounds of these projects here, emphasizing the points of intersection and divergence.

2.1.1 EWall

The EWall (Electronic Card Wall) Project, headed by Dr. Keel at MIT CSAIL, explored the application of information management and visualization algorithms to the construction of a software system for the support of individual and collaborative sense-making activities [Keel, 2004]. The project culminated in a software application based on William Pena's Problem Seeking methodology from the field of architecture and design, originally devised in 1977 [Pena and Parshall, 2001]. The methodology involves a "card wall" brainstorming session where ideas are rapidly generated and recorded on individual pieces of paper, which are then affixed to a shared wall or table. Participants freely rearrange and make changes to ideas, while continuing to generate new ones.

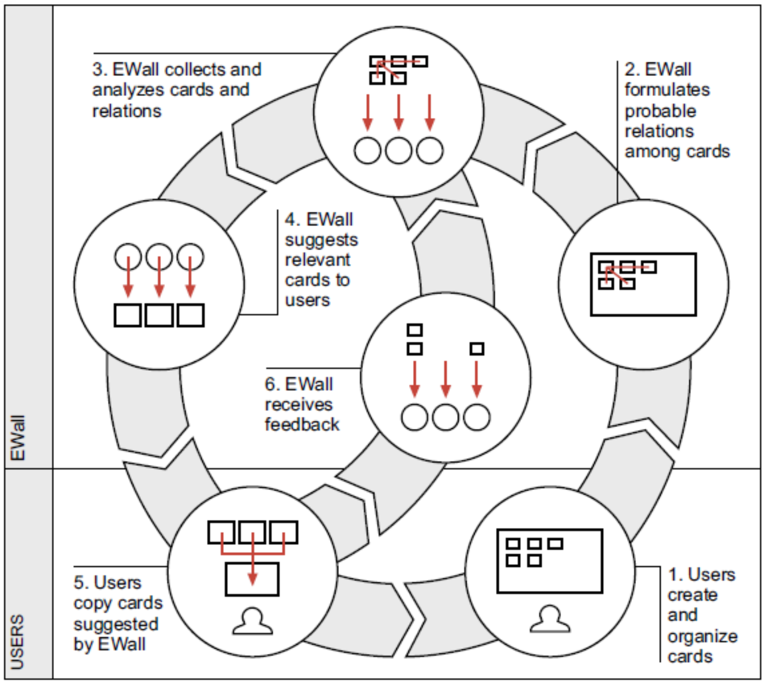


Figure 2-1: An illustration of the EWall system from Dr. Keel's Dissertation

EWall translates this behavior into a software environment where the system leverages computational techniques to create a rich interactive experience. Users organize electronic cards which represent discrete units of information. A network of software agents, each implementing a unique algorithm for inferring relationships or generating visualizations, work together to augment a problem solving session. The system was designed with the goal of catalyzing creativity and aiding in design and decision-making processes.

EWall had 3 main Technical Research Areas:

1. Information Visualization

2. Information Communication

3. Information Management

Several theses have been carried out within the scope of EWall. Agent systems have been studied for training information prioritization algorithms [Sither, 2006] and guiding decision making [Bevilacqua, 2004]. Visualization considerations were explored in great detail in [Li, 2004]. Computer deduction of spatial relations was investigated in [Kahan, 2004].

While the EWall Project serves as a launching point and provides many lessons, my thesis is differentiated in several key ways:

- Representation of workspaces based on content enabled by a **topic model**.
- A focus on **integration** with a variety of web data sources and tools.
- Establishing **persistent** problem space representations, or histories.

In particular, my thesis focuses more heavily on investigating relationships based on **content** as opposed to **context**. In EWall, the focus is largely on the spatial and temporal relationships among knowledge objects. Context is inferred by abstracting away the content. In my thesis, I implement a topic model representation and investigate content relationships based on this different type of abstraction.

2.1.2 Genesis Project

The Genesis Project, led by Professor Winston, is driven by the "Strong Story Hypothesis": Our intelligence spans from our ability to think in narratives, that is, to be able to represent our knowledge of the world, experiences, and other people through story-like representations. We are able to readily recognize both surface and structural relationships among stories, and constantly adapt our understanding to an incessant flux of knowledge from the world. Our ability to communicate rich, complex thoughts is made possible through the medium of the story.

At present, the Genesis system is capable of reading a simple text and identifying higher level plot elements. We say that it is capable of reflective reasoning. The system can answer natural language questions about events in the story, justifying answers based on its reflective understanding. The system operates by parsing a raw text into a machine-understandable form. The program then runs the parsed representation against common sense database and plot-element databases in order to identify plot units such as acts of revenge and pyrrhic victories.

The Genesis project and the present thesis share a common problem: Both need a higher level representation (abstraction) of complex documents to serve as the comparison layer. In the Genesis Project, the key to the machine's ability to make analogies among stories is the higher level representations superimposed on the workspaces. To understand the main idea, let's take a highly simplified view. Every story essentially has a topic vector generated for it, where the elements in the vector are plot-elements and themes present in the story. This forms the memory of the system. Then, if a new story is read by the system, it generates a new vector and then carries out a dot product against the vectors in its memory in order to determine which vectors are closest in distance and thus find candidate stories that may be analogous to the present story. From there, the system performs further analysis to identify deeper, structural relationships.

Similarly, in the Mental Model Web Browser software developed in this thesis, I use topic model representations to make comparisons among objects. However, these topic models are extended beyond just representing documents to representing entire problem solving sessions, as well as users.

2.1.3 AIRE Group

The Agent-based Intelligent Reactive Environments group [air], active around 1999-2002 and led by Dr. Howard Shrobe, features several projects that explore ideas related to those in my thesis. While I was not directly inspired by this group's work, interested readers may find the group's work to be a valuable resource.

2.2 Cognitive Science

In order to build systems that augment human problem solving, we need to understand the cognitive mechanisms that support our ability to solve problems. Research has been conducted into the cognitive mechanisms that give us our intelligence, but there is yet much to be learned. In this chapter I summarize the principles that have guided me in my explorations and developing the prototype system in this thesis.

2.2.1 Intelligence as a Function of Environment

Understanding and augmenting the human problem solving process is at the core of the present investigation. The pioneers in this field were Allen Newell and Herbert Simon. They studied the human problem solving process and created models and prototypes of some of the earlier software for solving problems. A key insight of theirs is packaged as "The Rationality Principle" [Newell and Simon, 1972] and claims:

> Much of the complexity of human behavior derives not from the complexity of the human himself (he is simply trying to achieve his goals), but from the complexity of the task environment in which the goal-sleeking is taking place.

This concept suggests that impact on the problem solving process can be made by adjusting the problem solving environment and catering to the physical plant of the human mind. In fact, models for human information processing have been quantified by Card [Card et al., 1986]. The way we interact with the world is governed by the fundamental physical parameters of our visual system, short term and long term memory, and cognitive processor.

The application developed in this thesis essentially creates an environment that captures the salient points of the user's thought processes and operates on them in order to augment problem solving. We explore the effect of controlling different elements of this environment and look for insight that a different and more powerful way of thinking is possible when the environment brings into focus an overall view of the knowledge-seeking session.

2.2.2 Extended Mind Hypothesis

More recently, David Kirsh has presented ideas on how our problem-solving mechanisms operate. In [Kirsch, 2009] he questions the rigid framework initially promoted by Newell and Simon and places emphasis on several key aspects of cognition. One of these is the interplay between internalized and externalized knowledge. Certain information is inherent and impossible to represent in an easily apprehensible way (processes, grammars, etc.) and remain largely relegated to an internal representation in our minds. Meanwhile, other types of information can be explicitly represented (facts, formulas, etc.), and can easily reside outside of our minds. Kirsh further points out that our problem-solving methods are deeply intertwined with environmental artifacts, that is access to information and tools that readily allow us to overcome questions and barriers as we solve problems. The present thesis aims to build an especially powerful tool in this spirit.

As the tools we develop continue to cater to our minds at in increasing level, the concept of *extended cognition* becomes more apparent. A recently survey [Smart et al., 2010] observes the augmentation of our cognitive abilities as a result of greater ubiquity of networked devices. There is yet much potential for sophisticated tools for externalize and retrieving knowledge in order to extend our problem solving abilities beyond the physical limitations of our brains.

2.2.3 Similarity is like Analogy

A computer system's ability to augment our intelligence requires it to help us make connections among data objects, discrete pieces of information such as documents, images, and programs. In order for the machine to make connections in a manner that we understand, we need a foundation for how it is we as humans infer relationships among objects. We derive a foundation for thinking about this problem from Gentner and Markman [Gentner and Markman, 1997]. Gentner analyzes the distinction between similarity as defined by surface features on one hand, and analogy as defined by structural alignment on the other. While we may more readily make connections and recall knowledge based on surface similarity, the comparisons that we find most valuable and informative are those where a strong structural relationship exists. For example, we might make a connection between an apple and a fire truck because they are both red, but the comparison essentially stops there. In contrast, Gentner draws an example from history involving Johannes Kepler, where the early astronomer made an analogy between the mysterious force that acts on large objects at a distance and the phenomenon of light, both forces whose affects are felt at origin and destination, but not as apparently in between. Gentner further points out that similarity based on surface attributes and analogy sharing little in common on the surface are two extremes of a continuous space. That is, similarity is like analogy, and both types of comparisons can converge upon literal similarity.

A collaboration between Forbus and Gentner [Forbus, 1995] resulted in a model of similarity-based retrieval, MAC/FAC, which provides further insight into the psychological phenomena behind similarity and analogy. A relevant part of their study showed that comparison of content vectors of data objects, or topic models, provides a computationally inexpensive way of predicting structural matches. Constructing and comparing topic models allows us to reduce the space of all possible matches to those that are good candidates for highly informative matches. In this thesis, I apply this idea to generating recommendations that assist user. The user is guided in making the knowledge connections he seeks in order to make progress toward achieving his problem-solving goals.

2.3 Minsky's Society of Mind

Marvin Minsky's ideas, collected in his book *The Society of Mind*, were some of the earliest inspiration for me and provide the nucleus of my own ideas and philosophy that I have been layering upon since first encountering Minsky about two years ago. I was fortunate enough to gain further understanding of his ideas through Marvin Minsky's seminar course at MIT during the spring semester of 2010. Here I describe some of these ideas and relate them to my research.

2.3.1 K-Lines Theory of Memory

In building a system that is an extension of the mind, one of the key considerations is how to design the memory. We need a model that resonates with the way our minds work, yet is not limited to physiological imitation but rather employs abstractions that may be represented computationally. Marvin Minsky's K-Line (Knowledge Line) theory of memory [Minsky, 1980] provides a model for forming and activating memories that I've found to be a great root for my explorations. The theory proposes a high level model for how a mind comprised of a society of agents, or resources, [Minsky, 1988], [Minsky, 2007] can be coordinated to yield a learning system. In Minsky's theory, a K-Line represents a cascade of the specific resources activated in response to solving a problem. Each resource may be activated by recognition of a specific feature in a problem, or by activation by connected resources. The K-Line records the network of connections among resources, or the mental state, which exists leading up to the solution of the problem. In the future, whenever a new problem is encountered, familiar features activate specific resources which then activate other resources on the same K-Line, eliciting a partial previous mental state. A superposition of partial mental states, combined with new information in the environment, leads to learning how to solve a new problem, and the recording of a new K-Line.

In the Mental Model Browser application developed in this thesis, the growing history of user browsing sessions is analogous to the network of knowledge formed in Minsky's model. That is, with every additional problem solving session, the corpus of previous sessions, or K-Lines, becomes greater and thus there are more opportunities for activation.

Activation and the Level Band Principle

Minsky's K-Lines theory explains how knowledge is recalled, or activated, through the Level Band principle. Activation happens when a stimulus from the outside causes a partial K-Line to fire. The question is how many resources should be activated. If we activate all parents, siblings, and children of the resource, we end up with a mess of activated resources and can't focus on any particular part to help us with the present problem. The chaos is alleviated by the Level Band principle: we activate resources an intermediate number of levels above and an intermediate number of levels below the point of stimulation. This results in a finite activation of ideas that are neither too general nor too specific to be useful. This "intermediate features" principle abounds in recent artificial intelligence and cognitive science research [Finlayson and Winston, 2006].

Goals

The notion of a goal in the context of knowledge-acquisition is a bit fuzzy and even Minsky provides a disclaimer about the impending confusion before proceeding to describe his notion of goals in the K-Line theory. In short, Minsky believes there is a mapping between a goal and the resulting knowledge that is gained through striving to achieve the goal. This mapping, however, is difficult to conceptualize as we don't always remember how we came to know what we do. For example, think about your ability to use a computer mouse; you'd be hard-pressed to recall what it was exactly you were doing when you learned how to use one. In the Mental Model Browser system the session-based workflow allows for the reification of connecting goals to knowledge. Every session inherently has associated with it a goal for that session. In fact the present experiment workflow requires the user to explicitly enter a goal (though this need not necessarily be the case in future iterations). The advantage of the computational environment is that we have the option of accessing knowledge according to the specific goal, whether or not that is the way we seek activation at a later time.

2.3.2 Frames

Another of Minsky's powerful ideas is the concept of frames [Minsky, 1974]. Minsky explains these as skeletons for knowledge objects, with slots, or terminals, to be filled by property settings. Frames serve the purpose of providing a schema of organization for related types of knowledge. There is a strong resonance between the concept of frames and object-oriented thinking prevalent throughout computer science. EWall uses the notion of frames for defining the attributes of different types of objects. The Genesis Project uses a family of frame-like representations. In this thesis, the topic model representation derives to some extent from the concept of frames. The concept vector representing each document is a sort of frame, with slots for topic words. Frames are intended to provide a medium for comparison among objects in the world, much as the topic models serve this purpose in the Mental Model Browser.

2.4 Web Technology

2.4.1 Delicious - Topic Models from the Crowd

Delicious [del] is a popular web service that allows users to simultaneously save and publicly share their web page bookmarks. Further, users are encouraged to 'tag' the web pages that that they bookmark, that is provide labels that are meaningful to the user to describe the content of the page. The tags then serve two purposes:

1. as the user's interface for searching his or her previous bookmarks by topic.

2. as a platform for public discovery of resources by topic.

For example, when doing research on "artificial intelligence", a user would tag every bookmark that he creates with an identifier such as "ai" or "artificialintelligence", along with any other relevant tags. He can then search his own bookmarks tagged with this label, or all public bookmarks across Delicious, providing a means of discovering new resources on artificial intelligence. Meanwhile, he has contributed to a valuable corpus for others to use for discovery. This results in the organization of millions of web documents according to multiple dimensions as defined by these topic tags. Every bookmark provides a dimension or means of filtering. Figure 2-2 gives a screenshot of Delicious, illustrating URLs that have been tagged with "ai" and "ui".

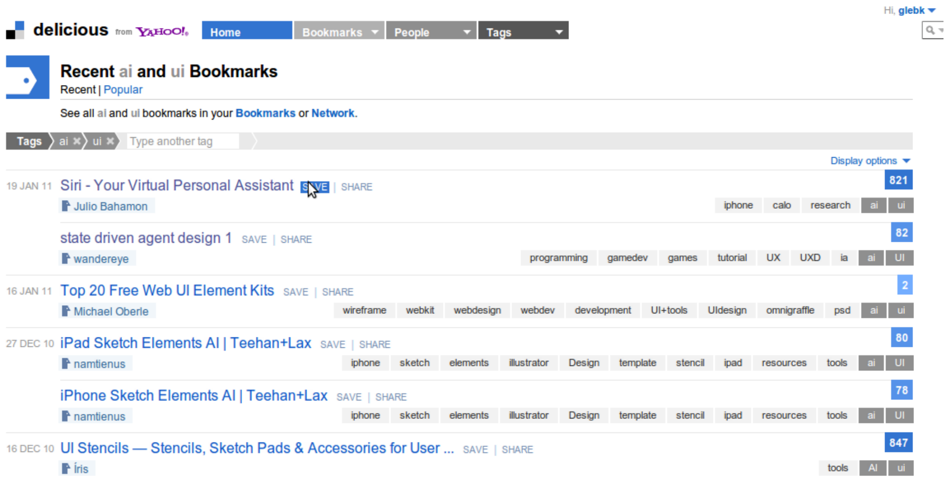


Figure 2-2: Delicious query of URLs that have in common the "ai" and "ui" tags

Note that each document may also have additional tags which allow for discovery of those documents in different topic planes.

For our purposes, Delicious can be used as a good starting point for a recommendation engine for several reasons:

Approximation of Topic Models

An emergent result of the Delicious bookmarking service is that when a page is tagged sufficiently often, some common tags emerge which effectively represent a fair approximation of a topic model of the page.

Tag-based Querying of URLs

Delicious allows querying by one or more tag. For example, we can query according to the tag "ai" to get all instances of pages that have been bookmarked and tagged with this label by an assortment of users. We can further refine our search by tagging a page with "ai" and "ui" to get the URLs that have been tagged with both of these. Clearly the more tags one uses the more refined the results; however, after about three tags the number of results returned diminishes quickly, though the relevance may increase drastically.

Quality Filter

The Delicious API only exposes pages that have achieved some critical mass of attention. The result is a quality filter that guarantees that the pages returned were found valuable by several individuals, and reduces noise and spam.

Delicious serves as the data corpus for the intelligent component of the system developed in this thesis. The features of Delicious outlined above make it an excellent starting point for an intelligent recommendation system. Further, the layers added by the Mental Model Browser application may inspire more users to augment their workflows with intelligent recommendation and potentially create a source of richer crowd-sourced data.

Chapter 3

Mental Model Browser

The Mental Model Browser is an application that provides a layer on top of the typical web browsing experience. The application is intended to be used on a per-session basis. The user comes up with a specific question and then launches the Mental Model Browser to run in parallel to the browsing session. The software records the URLs that the user visits, constructs a representation of the overall session, and provides real-time recommendations throughout the session. The session is then saved in a history database which serves as a future resource for the user. Future development iterations will allow for the history to be used as the computer's source of knowledge about the user as well as a shared resource for collaboration which can benefit other users.

3.1 Philosophy and Central Ideas

The motivation for carrying out this thesis research and the source of design decisions I've made in implementation span from a distinct philosophy that I've derived through background reading, conversations, and constructing prototypes of related tools for the past two years. The system developed for the thesis is in effect a medium for testing the ideas in the philosophy. Further, the philosophy provides a roadmap for future work.

3.1.1 Everything has a Topic Model

In EWall, the unstated mantra is "Everything is an object" (though Dr. Keel never forgets to mention this in our conversations). In order to transition toward a content-driven system, we apply the mantra "Everything has a topic model".

When dealing with dense pieces of data, such as a web document, it is necessary to come up with an abstraction that captures the main idea in the web document, yet is compact enough to be amenable to computational manipulation. We seek an efficient representation that allows for the ready comparison with other documents. For example, if you were handed a book and told to analyze it, it would essentially require you to read a good portion of it in order to determine what it was about. A computer algorithm that processes entire documents for pairwise comparisons would be slow and inefficient. A simple bag of words analysis is often insufficient to extract the most salient concepts, as the frequency of the words' appearance in the text, if it appears at all, may undervalue its importance.

Building on well-known ideas in information retrieval, I utilize the notion of a topic model, a statistical model that exposes an abstract representation of a target data object, or resource, as a set of concepts representing the contents of the resource. In the Web domain, the word **tag** describes a label given to a web page or other resource (image, product, etc.), and represents the notion of a topic. The abstract representation typically comes in the form of a concept (tag) vector which describes the content of a resource in terms of a handful of concept words (Figure 3-1).

The motivation for using a topic model representation is the potential for making on-the-fly comparisons in a manner that is not computationally overwhelming yet allows determining to what extent two data objects are related. Further, topic models facilitate comparison among data objects of different nature. For example, an image can be compared with a text document, or a web document can be compared with the transcript of a conversation. A document may even be compared to a collection of documents. This last idea is critical to grasp in order to understand how documents will be retrieved to match the context of a web browsing session (Figure 3-2).

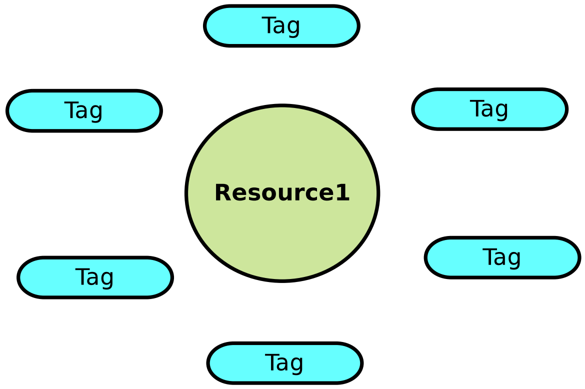


Figure 3-1: Resource Topic Model as a Collection of Tags

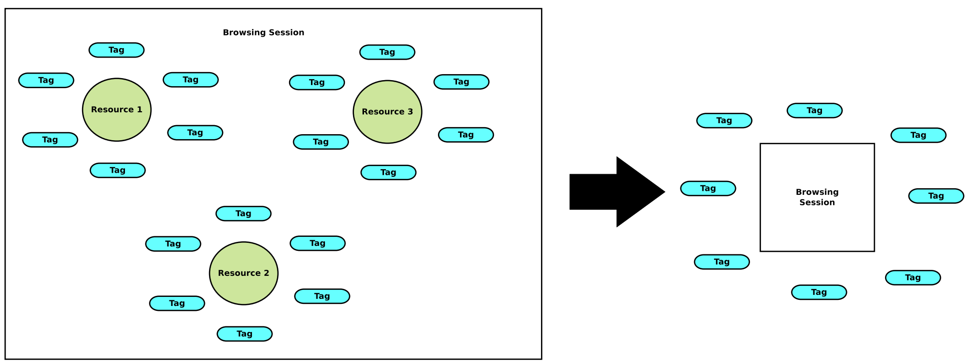


Figure 3-2: Session Topic Model

Gerard Salton originally pioneered the use of concept vectors in information retrieval and Forbus and Gentner [Forbus, 1995] illustrated a model for similarity discovery which uses concept vectors for predicting structural alignment. By comparing vectors in a manner analogous to finding the cosine angle between two vectors, it is possible to predict the structural similarity between two objects. This is computationally implementable and thus can filter a large space of comparisons and present it to the user for deeper analysis.

Finally, a consequence of the assumption that we can form topic models on different hierarchical levels is that the user himself can be represented as a topic model. More precisely, the user's present mental state may be seen as a topic model of the ideas that are in his working memory. Further each topic links to knowledge in a manner very much in the spirit of Minsky's K-Lines, as illustrated in Figure 3-3. Thus in the anteroom of the consciousness there are always related ideas that are activated by the present memory and the external stimulus of sensory input and knowledge in the world.

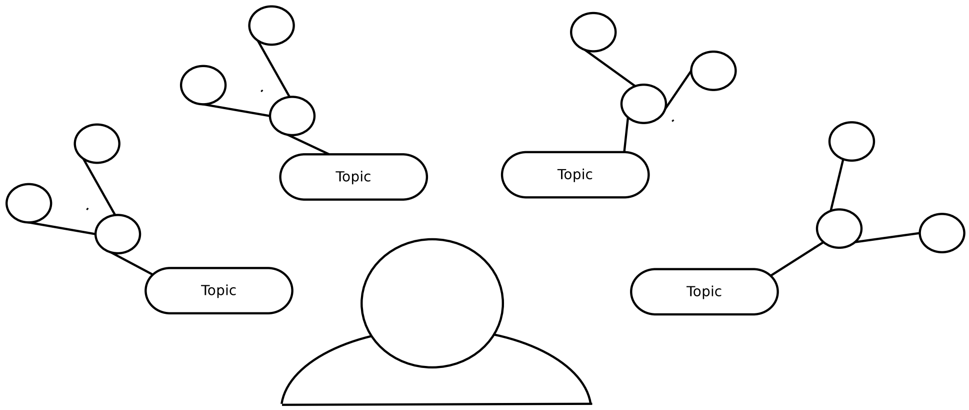


Figure 3-3: The User's Mental State as a Topic Model

There are several implications of a computational representation of the user's present mental state.

- In this thesis, having a model of the user's mental state allows recommendations to be made that better cater to the user, not just to the current session. For example, a user who is an expert physicist and who conducts a search for a mathematical topic should not get the same response as a user that is a beginning student. The system distinguishes between the backgrounds of these users in order to provide respectively appropriate results, even if the goal of their sessions appears the same at a certain moment. The expert physicist will be interested in recommendations that match his deeper knowledge of the topic, while the student may appreciate something more basic.

- Every topic in the working memory of the human is tied to previous knowledge, perhaps via a mechanisms not unlike that of Minksy's K-Lines. This means that surface activation may complement the ideas that are already on the threshold of consciousness and yield a slightly different mix for deriving structural relations that relying on the human memory alone would miss.

3.1.2 Tagging and Meta Data

While the actual text of an online document can be analyzed to paint some picture of what the document is about, a different type of understanding may be gained by exploiting tag and meta data about information sources. Tags and meta data can be created either by the producers or consumers of content, effectively establishing directed edges from individuals to resources, labeled according to the tag. I see this as a starting point for making the mapping between *people* and *information* more explicit. Further, tagging exposes the relationship between *information* and *information*. And as a result, relationships between *people* and *people* are exposed as well. The tagging paradigm is widely employed but its full potential is not yet realized. Delicious [del] provides a great data corpus for tagged web pages and provides a means of approximating topic models in the Mental Model Browser application.

3.2 Design of the Application

The space of features that augment knowledge exploration is exponential in size and it was critical to choose only a few select features to implement in order to remain within the scope of this thesis. Further, with every feature, it was necessary to make some critical design decisions. Each decision was guided by a mix of intellectual curiosity to address the philosophical explorations mentioned in the previous chapter, while adhering to the simplest solution whenever possible. Of course when experimenting with new interfaces, Occam's razor tends to get a bit dulled, but without risk there cannot be reward. Figure 3-4 gives a high-level snapshot of the system.

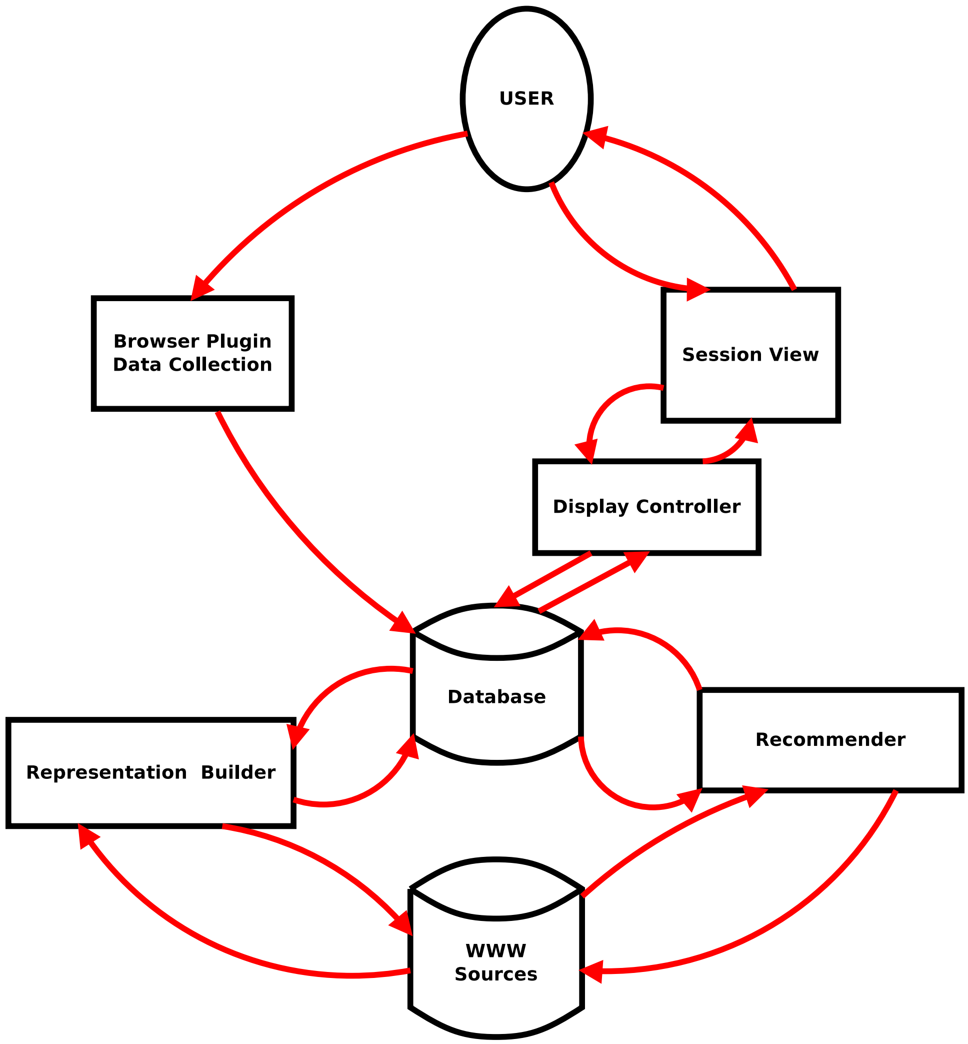


Figure 3-4: System Overview

3.2.1 User Interface

Designing a user interface requires a highly iterative process as it is difficult to predict exactly how a user will use the interface, especially in an application of this nature. I went through several iterations before I finally settled on the final design illustrated in Figure 3-5. Given the horizon of the experiment, it was critical to pick the minimum number of features that would make the system usable, yet avoid becoming too complex and overwhelming. One of the important issues to consider was which regions were most important to present to the user. The main features chosen were:

Recording Toggle Button - enables the user to start and stop recording.

Submit Session Button - allows the user to mark a session as one worthy of analysis in the context of the experiment.

Session Goal Input - requires the user to briefly describe what the goal of the browsing session was.

User Tags - the user adds tags to describe the salient concepts present in the current browsing session.

User-visited URLs - a record of the URLs visited during this session.

Recommended Tags - dynamically generated and continuously updated inference of the present session topic model based on the visited URLs.

Recommended URLs - URLs retrieved based on the current session topic model.

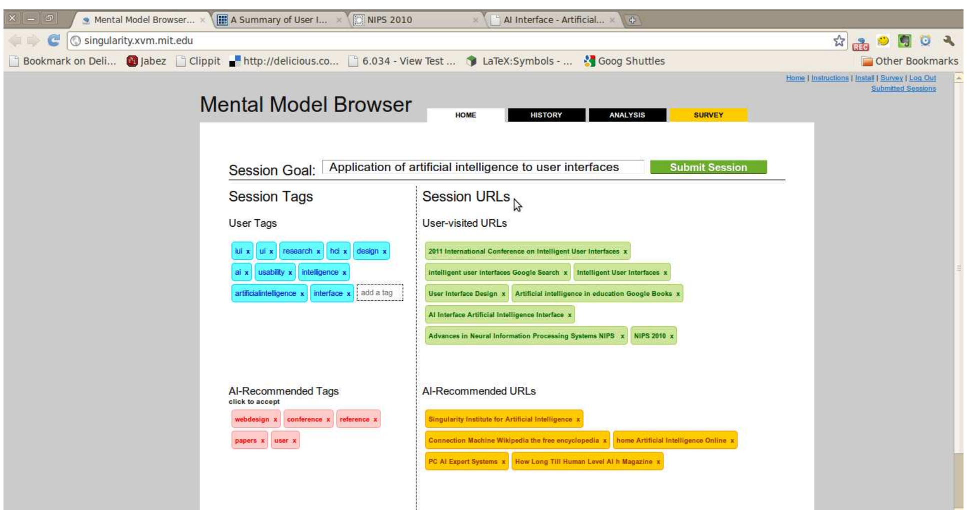


Figure 3-5: Mental Model Browser Main Interface

3.2.2 Session-Based Browsing

With the goal of this thesis being to capture and augment problem solving sessions, I chose to require the user to indicate when they were beginning a session and when the session was complete. In order to convey this idea, the version of the application in

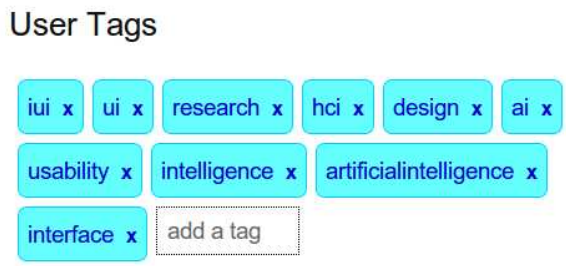


Figure 3-6: User Tags

Figure 3-7: User-visited URLs

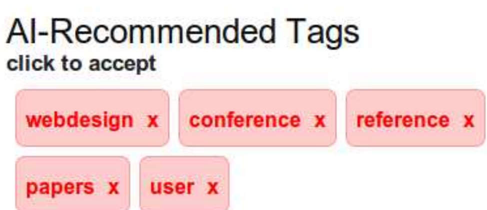


Figure 3-8: AI-Recommended Tags

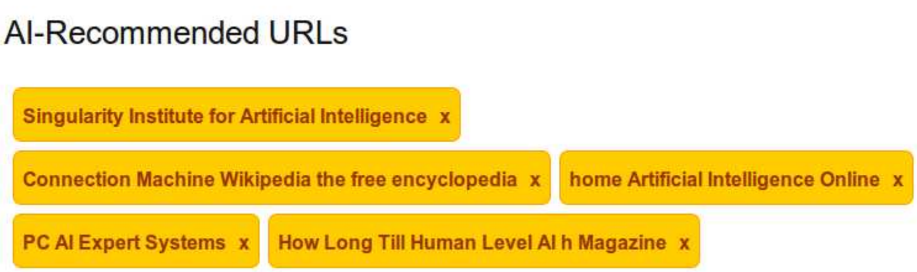


Figure 3-9: AI-Recommended URLS

this thesis was structured and described as intended for "record a browsing session". That is, there is a specific beginning and end to every session.

The alternative extreme would have been to give users complete free reign and impose the task of recognizing context switching onto the machine. The application would run continuously as long as it was installed. However, at this stage, it would have been a tough sell to convince my users to give up more privacy than I was already asking, and so enabling the user when to start and stop sessions was a good choice for the version of the application tested in this thesis.

The per-session design additionally enables a more clear connection to the ideas explored in both the EWall and Genesis projects. In EWall, the collaborative sessions are structured around solving a specific problem. Users have a workspace for the present project being solved. The system makes inferences based on this workspace and provides recommendations to collaborators and the user him or her self based on the data in the workspace.

The connection to Genesis is that the per-session discretization adds a specific finiteness to the object for which we construct a topic model, or concept vector. That is, the session as a whole gets a topic model and recommendations can be made by comparing the session concept vector with those available in online sources and the system history.

3.2.3 Data Capture

Design Dilemma

In a system that seeks to augment the human problem solving process, there is a fundamental question about the level of invasiveness that the data capture process brings in. One of the recurring themes in my conversations with Dr. Keel was how much constraint the system should put on the user. If the system has too many constraints on the user, that is requiring the user to adapt to a process that is too different from his or her normal behavior, then creativity will be compromised and the likely result is that the user will stop using the system. On the other hand, if there are not enough constraints, the user will not be able to figure out a way to use the system.

The question often seems to come down to a dichotomy between whether the user should be in control of every data object that is sent to the system or whether the system should have complete control and responsibility over what data it chooses to collect. The precedents of web browsing technology usually pick one of the two extremes. For example, the default web history setting in the majority of web browser is to record every website that is visited. This creates a stream of information that can later be leveraged for form input completion/suggestion or visited-link indication. However this approach misses critical input from the user about what is important and why the user went to those websites.

On the other extreme is the paradigm of the browser "bookmarklet", a button that the user installs in order to provide an active interface into the application. If the user desires to take the action provided by the application, he or she must actively press the bookmarklet. This is common with Bookmarking applications. The problem with this approach alone is that the user does not always think to create a bookmark so critical points of the browsing history can be missed.

Solution: Hybrid Passive and Active Approach

The resulting design features a combination of passive and active recording. The recording control is described in Figure 3-10. When the user decides to begin a browsing session, he indicates this to the application by pressing the record button installed next to the address bar of the Google Chrome browser. To finish recording the user can either press the recording button again, or press the Submit Session button in the session view (main interface). In between, the system passively records the URL of every web page visited and sends it to the server which writes the information to the database.

For the purposes of the experiment, the session was not submitted for evaluation unless the subject wished to submit the session. This limited submitted sessions to those that were intentionally submitted and prevented users from submitting a session

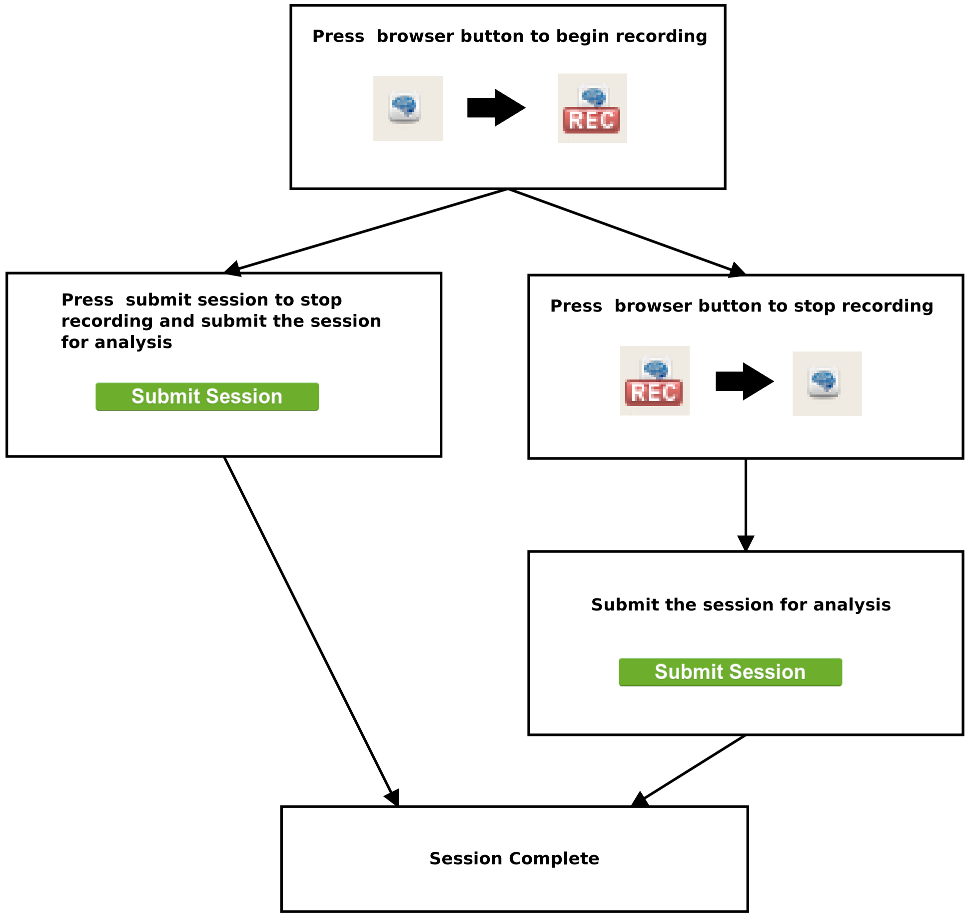


Figure 3-10: Data Capture Flow

that was incomplete or lacked substance.

As the session progresses, a session view is updated with the resources visited by the user (Figure 3-7). Before submitting the user can choose to delete the resources that were irrelevant to the session by pressing the **x** button on the resource object (present on all the data objects).

3.2.4 Recommendations

The recommendation system described here represents the “intelligence augmentation” stage in the Mental Model Browser. There are no complex algorithms for accomplishing this in this thesis, as the purpose of this thesis was not to explore complex algorithms, but rather to design and implement a process loop that epito-

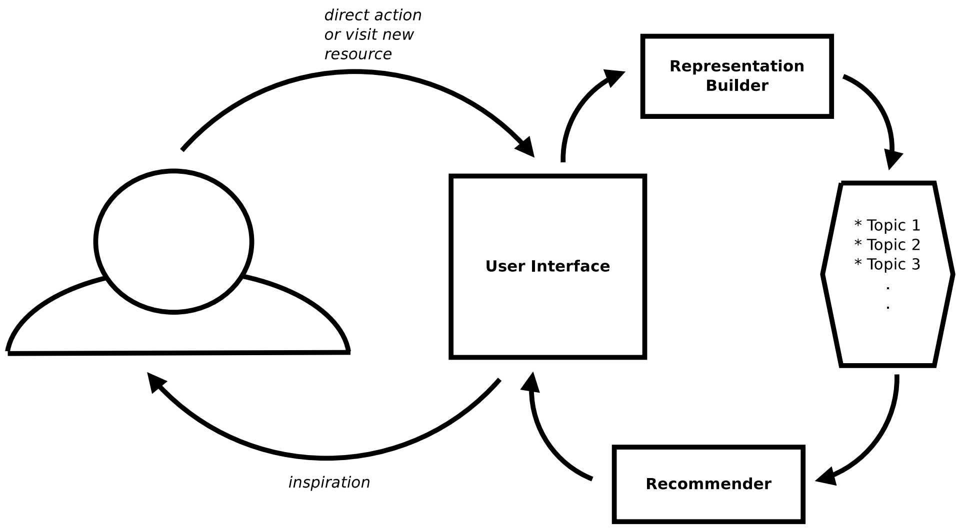


Figure 3-11: Recommendation Loop

mizes the exchange of information between the machine and the human. Figure 3-11 illustrates this design.

The interface uses two types of recommendations are presented:

Session Tags

These provide the means for the system to get feedback from the user about whether its understanding of the session is correct. In other words, this is the system's "best guess" at what additional features the topic model of the session might have. If the user accepts any of these recommended tags, then they are heavily weighted toward the actual session topic model.

Session URLs

The recommended URLs come from the present topic model of the session. That is, the topic model is essentially a dynamic query that is used to draw the user's attention to potentially related sources.

As the user is browsing, he provides information to the system through the resources he visits or through direct actions on the interface (e.g. adding tags, deleting irrelevant resources and recommendations). The system uses this information to continuously update its topic model of the session and update the recommended resources. The recommendations are largely based on surface similarity [Gentner and Markman, 1997]. In future work on the system, however, it may be possible to look for structural matches if a greater corpus of previous usage data is available.

3.2.5 History and Analysis

For now the history and analysis views are largely for informational purposes and are not very functional. They were designed to allow me to assess how users were using the system, as well as provide users with after-the-fact feedback. The intent was that users would observe the result of their session and potentially come up with a better understanding of what the system is intended to be used for, as well as come up with ideas for how the system could be used.

History View (Figure 3-12)

The history view provides a list of previous sessions. Eventually this could be searchable, acting as a more powerful version of the current Internet history. In fact most people don't presently use their history for any practical purpose. This type of interface, however, is designed around user goals and could be useful if one remembers doing a particular search and wants to see the results of that session.

Submitted Session View (Figure 3-13)

This is a snapshot of the tags and URLs that the user chose to include during the session.

Time-Step Session View (Figure 3-14)

This view arranges the session objects (user tags, user URLs, recommended tags, recommended URLs) in order of creation so that an observer may get an idea of how the session came about.

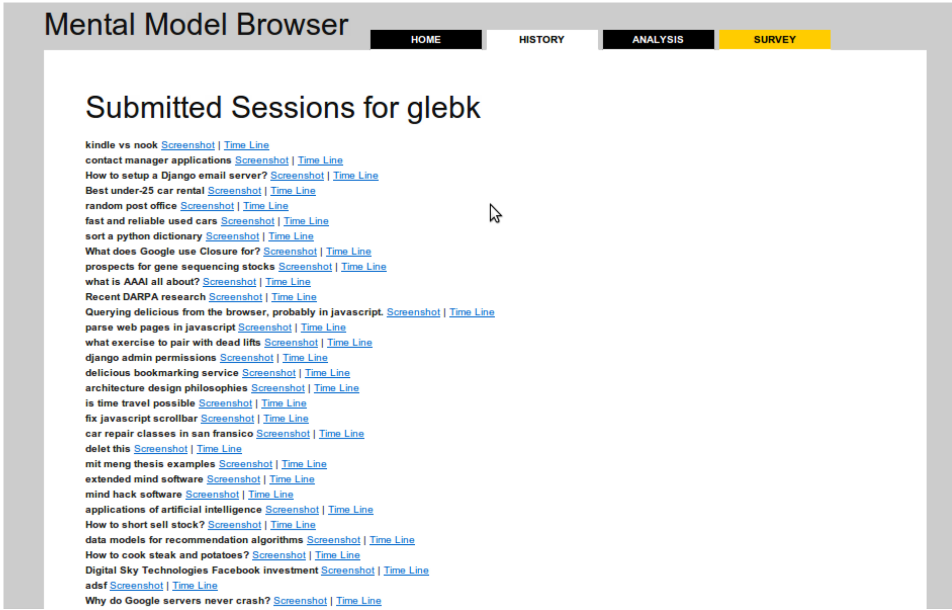


Figure 3-12: History View

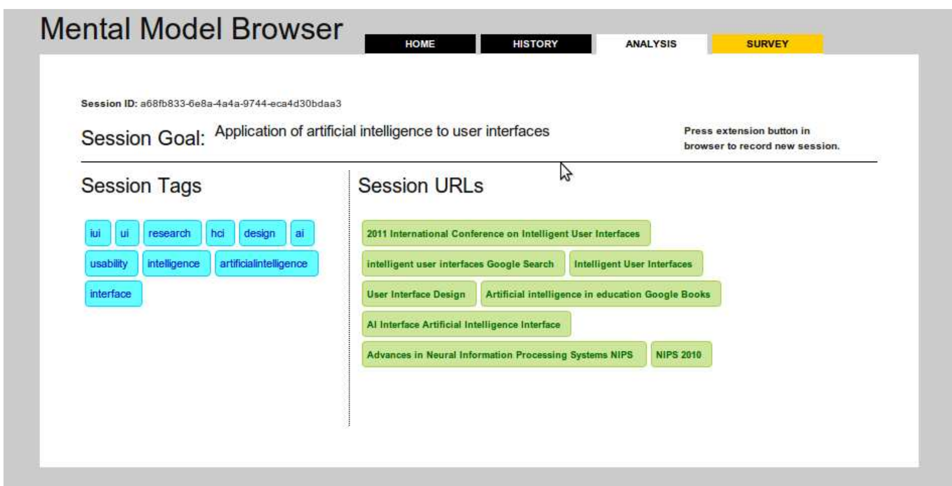


Figure 3-13: Previous Session View

3.3 Implementation

3.3.1 Data Model

The data model of the application is reflected in the interface. The core objects are illustrated schematically in Figure 3-15.

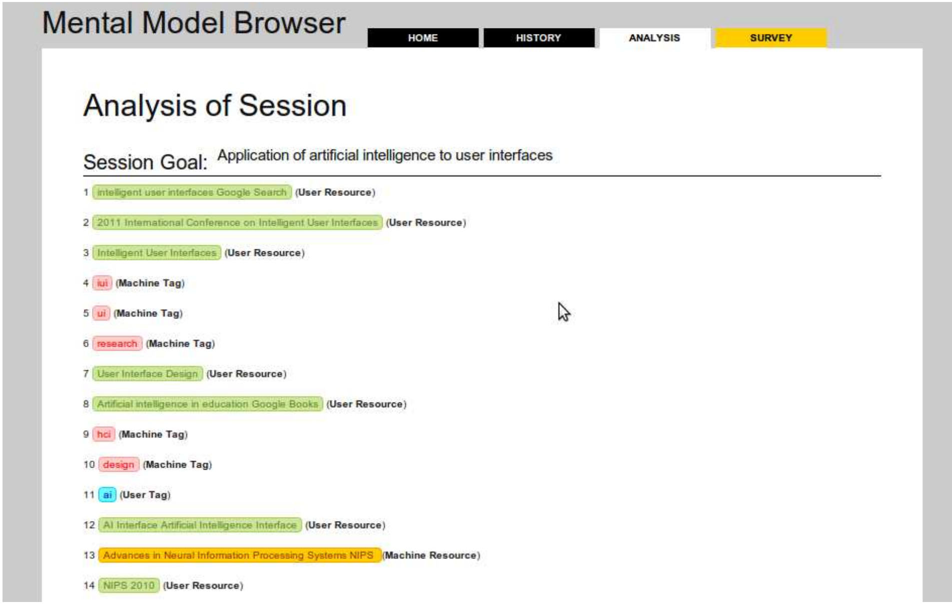


Figure 3-14: Time Analysis View

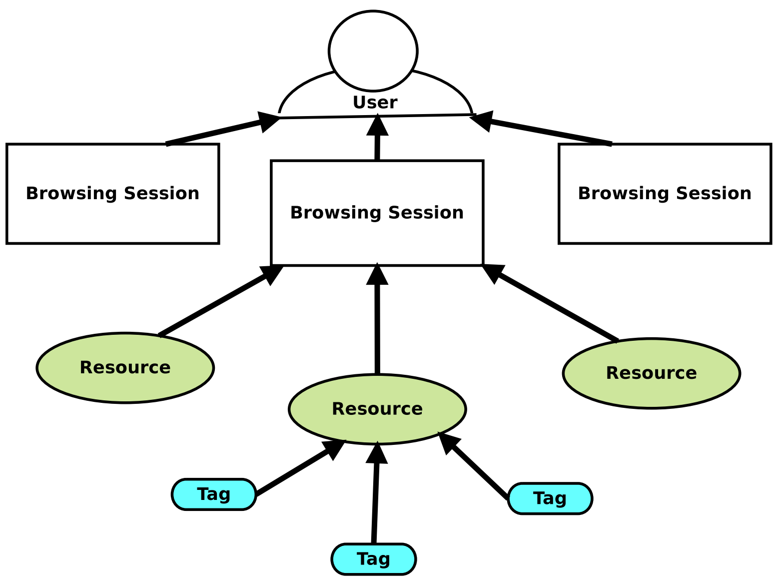


Figure 3-15: Core Data Model

3.3.2 Recommendation Algorithms

The recommendation feature of the application was implemented by clever interfacing with the Delicious bookmarking service API.

I implemented a simple algorithm for coming up with the present session topic model. In words, it takes the user chosen tags as the authoritative topic model if any have been included, otherwise it takes the machine inferred tags.

Build-Session-Topic-Model(session-ID)

```
sessionTags ← Get-Session-User-Tags()
if len(sessionTags) == 0 then
    sessionTags ← Get-Session-Machine-Tags()
end if
return sessionTags
```

The algorithm for getting recommendations picks a subset of the sessionTags and queries the Delicious API for matching URLs.

I developed and tweaked two algorithms for recommendations, one for querying the tags associated with with a newly visited resource. (This one was a straight-forward call to the Delicious API). The other algorithm provides recommendations based on the present topic model of the session. This also was implemented as a demonstration of one possibility:

Get-Recommended-URLs(session-ID)

```
recommendedURLs ← [ ]
sessionTags ← Get-Session-User-Tags(session-ID)
while len(recommendedURLs) == 0 do
    tagsSubset ← Random-Subset(sessionTags)
    recommendedURLs ← Query-Delicious-API(tagsSubset)
end while
return recommendedURLs
```

where Random-Subset() just returns a subset of 2 or 3 (random) of all possible tags; with repetitions possible. [1]

One issue that I faced here, a typical consideration when interfacing with an API, was a limitation on frequency of access as defined by the terms set down by Delicious.

[1]The randomization was chosen in order to force some variety into the recommended URLs for a session. This was desirable in order to increase the chances that users would find something useful and choose it. Of course this would come at the expense of plenty of irrelevant recommendations.

A lot of public data services place limits on how often they can be queried. The limit imposed by Delicious is it allows querying once per second. (In fact, while working out the bugs in the system, I was throttled on numerous occasions and would have to wait for a few hours to resume activity.)

3.3.3 Web Infrastructure

Django Web Framework

A complete web stack forms the basis of the application. The core was built upon upon the abstractions provided by Django, an open source web framework written in the Python programming language. My choice of this framework as opposed to others was to reduce development time and allow integration with intelligence algorithms written in Python.

Browser Extension

I found the browser extension to be the best way to collect data for an experiment of this nature. A browser extension allows JavaScript code to be injected into any page that the user visits, which I used for sending the URL and title of the page to the server. I built an earlier extension for the Firefox Web Browser but found among my early users that most used Google Chrome. Upon switching to the Google Chrome extension development framework, I realized that it was much better development environment.

I also debated the level of passiveness of the extension. Due to privacy concerns, I decided to make the extension only function when the user pushes a button installed next to the address bar. This would ensure that unintended data was not sent to my server.

Finally, the Google Chrome extension turns out to be a great decision in laying a groundwork for future development, as I will be able to display recommendations and results directly in the browser without requiring the user to jump back to the homepage of the website. Further, the Google Chrome extension development environment allows for using the browser's local storage so that it can operate independently of a server, ensuring a higher level of privacy for those users that desire it.

AJAX Data-Handling

The use of AJAX (Asynchronous JavaScript and XML) was critical to the usability of the application. Especially in one of this nature where there were many working parts. In particular, with recommendations coming in and actions being sent back to the server, it was key not to have a screen that kept refreshing or being disruptive in other unnecessary ways. One huge enabling feature of AJAX is refreshing sub-parts of the screen. This results in a seamless experience for the user and allows different parts of the same view to operate independently.

AJAX techniques also allowed for content to be displayed to the user in a smooth manner. The server would communicate data in the form of object arrays, and front-end JavaScript code would arrange the data into visually appealing boxes.

3.3.4 Server Administration

Full control over the server was a necessary evil that definitely came in handy in implementation of the Mental Model Browser. I was able to borrow a slice of a virtual machine from `http://xvm.mit.edu`. Here I installed an image of Ubuntu 9.10. By controlling the server I was able to administer a MySQL database, as well as install a queue server RabbitMQ in combination with Celery which was used for asynchronous representation-building on the back-end, as well as communication with the Delicious API. I also set up the combination of Apache and Nginx to serve my site, and maintain logs (which I thankfully never had to dig through as the server miraculously did not throw problems throughout the duration of the experiment!)

3.4 Early Prototypes and Experimentation

Before the Mental Model Browser, there were a few applications that I built that were my first forays into tying together the visual workspace ideas in EWall with the similarity and structure discovery ideas in Genesis. Each of these taught me some lessons which I was able to apply in developing the present prototype.

3.4.1 K-Lines Web Browser Application

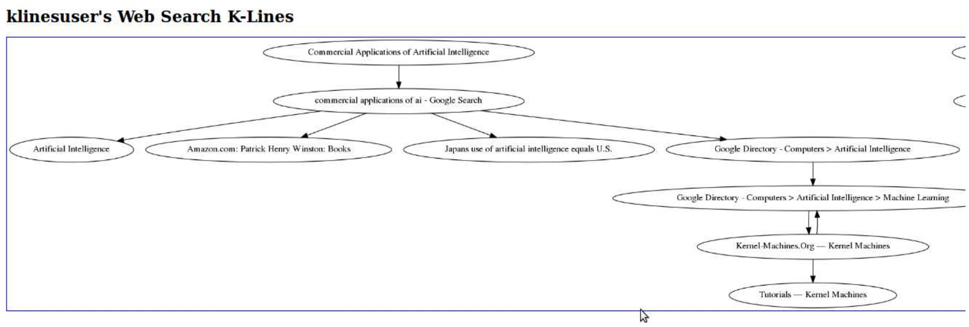


Figure 3-16: K-Lines Web Browser

The K-Lines web browser application was my first attempt at implementing an innovative tool for visualizing web browsing behavior. The idea was to record the user's browsing session and dynamically generate a graph showing the sequence of resources as he was trying to solve a goal - very similar to some of the ideas explored in the Mental Model Browser. A history would be generated over time so that whenever a user visited a resource he had visited before, the previous K-Line would be shown to indicate there is overlap with a previous problem.

After some initial user testing, I decided to set aside the ideas in endeavor as there was too much dependence on reaching a critical mass of data before the application became useful. In fact, this realization led me to focus on generating recommendations in the Mental Model Browser application so that the user received immediate utility from using the application. As the critical mass issue gets addressed in the future, the ideas about activation from this project may be reincarnated as users begin to not only get recommendations from the web, but from a representation of their own history, or external memory.

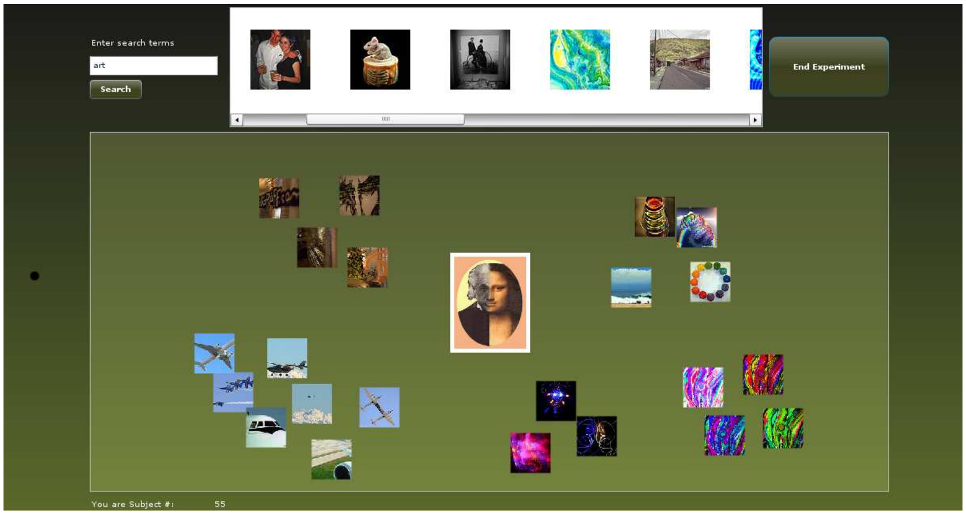


Figure 3-17: Cogwall Visual/Semantic Knowledge Experiment

3.4.2 CogWall Visual/Semantic Knowledge Experiment

The CogWall (Cognitive Object Wall) experiment, playfully deriving its name from EWall of course, was an early attempt at investigating the relationship between human spatial knowledge and content knowledge. Users were asked to search for images and generate arrangements representing several different concepts. The images came from the Flickr database and had user-generated tag data associated with them. Just as in the present thesis, this tag data was used to approximate topic models. The study involved identifying concept clusters using k-means clustering on physical locations and comparing the spatial clusters to clusters generated by according to content vector distances. A basic recommender feature was added that allowed a user to search a concept and have a representational image automatically placed in the workspace at a location associated with the correct concept being represented.

The project taught me just how many degrees of freedom spatial arrangement of objects enabled for a user - concepts investigated to a deep extent in EWall. I realized that there was a lot to learn on the content side and would later further investigate the concept vector idea more thoroughly in the present thesis.

Chapter 4

Pilot Study

4.1 Purpose

The experiment was conducted in order to evaluate how the application was used, its self-reported utility to the users, its usability, and to gain insight into how users adapted to the framework, or ways of thinking inspired by the framework. The results provide guidance for future design iterations, as well as insight into the relationship between internalized and externalized knowledge.

4.2 Method

Users were given a link to the experiment website where they were given instructions on installing the web browser extension and instructed to record several browsing sessions. (Complete instructions in Appendix 5.3.2.) The instructions included a request to complete a brief survey about their experience which had the following questions:

1. How useful were the tag recommendations? (1 = not useful, 10 = very useful)
2. How useful were the resource recommendations? (1 = not useful, 10 = very useful)
3. Please explain above ratings.

4. Rate the usability of the overall application? Was it cumbersome? Was it useful? (1 = too hard, 10 = amazing)

5. Please explain above ratings.

6. Do you have any suggestions for improving the application? Or ideas for features that would motivate you to use such an application more?

4.3 Analysis of Results

The experiment saw 22 participants submit 56 sessions. The participants were an approximately even distribution of MIT undergraduates, graduates, and young professionals. Most had a background of computer science. The sessions submitted by the author were not included in the analysis. (*Screenshots of several examples sessions in Appendix B*)

4.3.1 Usage Statistics

Participants	22
Sessions Submitted	56
Sessions with at least 1 User Tag	35
Sessions with at least 1 Recommended Tag accepted	24

Table 4.1: Overall Usage Statistics

Looking through the submitted sessions, I noted that several did not have any User Tags. I filtered out those sessions and ran the statistics again to get the data in Figure 4-1. All of the numbers went up, with an especially significant jump in User Tags and Recommended Tags.

I also noted that not all sessions had accepted at least one Recommended Tag, so I applied that filter to get the numbers in Figure 4-2. Again the numbers went even further up. Further, the proportion of Recommended Tags as compared to all User Tags was significantly greater.

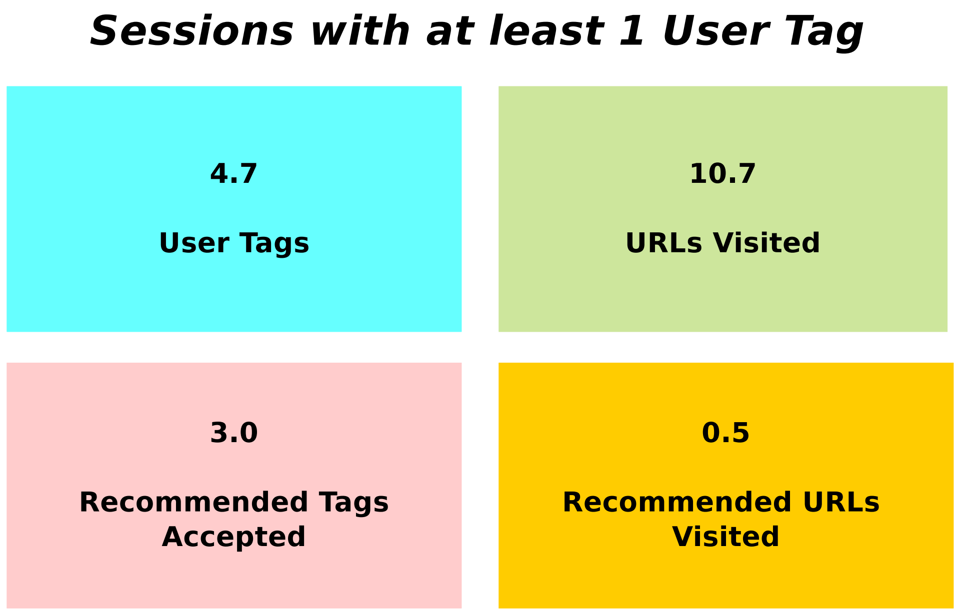


Figure 4-1: Sessions with at least one User Tag

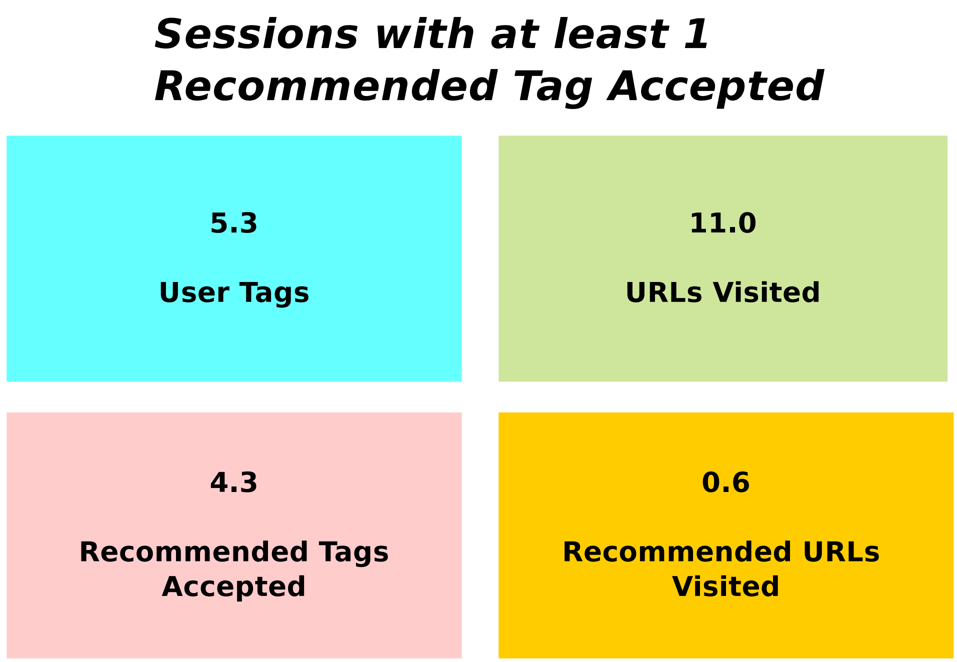


Figure 4-2: Sessions with at least one Recommended Tag Accepted

Among the users who used any tags at all, those who used recommended tags ended up with more tags and slightly more recommended URLs. This implies that the tag recommendations gave the users more opportunities for externalizing their intent. And the greater number of user tags may have resulted in better URL recommendations; hence the observed increase in Recommended URLs visited.

4.3.2 Survey Results

Number of Survey Respondents	10
Average Tag recommendation rating (1-10)	6.3
Averaged URL recommendation rating	6.3
Average Usability rating	6.5

Table 4.2: Survey Results

Selected User Observations

"The recommendations seem to be getting better from day to day."

In fact nothing was changed in the algorithms since this user began using the system. The phenomenon that may be happening is that the user is getting accustomed to how the system works and is actually adapting his way of thinking in order to better leverage the system - adding more user tags, etc. For comparison in the realm of typical web search, there are certain techniques that yield better results when running a Google search. Users generally figure out through trial and error that using more content words and avoiding vague words leads to better results.

"The recommended tags were spot on but the recommended URLs were not that useful"

It is easier to come up with a topic model of a document, then to take a topic model and come up with a relevant resource. User feedback showed that recommended URLs were pretty hit or miss.

"Tag Recommendations for the most part echoed words in submitted URLs. Some of the URL recommendations were valid, however."

This user found the tag recommendations to be obvious. This is a good sign in the *sense that the recommendations should follow from the session, and generally should echo words observed in the session. The fact that the user observed valid URL recommendations means that he or she was able to adapt to the tag paradigm and aid the system in constructing a model of the session.*

"The recommended tags were usually spot on until I selected a few (3-5) of them as correct, at which points their replacements became rapidly less relevant."

The observation by this user matches, in combination with the statistic in the usage numbers, highlights that 3-5 tags per session has some significance as a figure. It may be that more tags create too much noise, while any less is not descriptive enough.

"The application was very easy to use, I have a few small suggestions below. The only reason it gets an 8 is that it's hard to remember to go back to the tab with all of the information as I browse (although I was happily surprised to find that I quickly became accustomed to go to that tab when I was stuck on my search). Basically having some kind of non-intrusive sidebar or toolbar would make it smoother."

This user gives good evidence that the non-traditional flow established by this application can be adapted to. He was initially inconvenienced by having to jump back to the home screen tab, but actually found that it helped him get past points where he wasn't sure what else to search for. By providing an overview of the session so far, the user can reinforce their focus about what they were trying to accomplish and think of a next step. The recommendations may also assist in inspiring next steps.

4.4 Discussion

4.4.1 Effect of Tags

The tags were designed to be the main component of the dialogue between human and machine. As the server received more resources from the user during a session, the system derived its best guess at meaningful concept words and displayed them to the user. Whatever tags the user entered were taken as the authority for the meaning of the session.

The data shows that in sessions where users actually added tags, more resources were visited. Further, in the sessions where at least one machine tag was accepted, there was a greater number of tags and resources, and a higher proportion of the user tags came from machine recommendation. This suggests that the system helped users frame the problem they were going after and by eliciting feedback, kept the user engaged for a longer period. This sentiment is confirmed by some of the written survey answers.

The tags also became the mechanism for users to tweak the value of the recommendations they were receiving. Whether or not it was a fair question, the average ratings of tags and URLs turned out to be the same, even though the common sentiment was that the tags were usually pretty relevant, while the URLs were only sometimes relevant. Users showed quick adaptation to the tag model and more readily became comfortable with the use of tags.

I propose adding a history component, where users can search for previous sessions by tag, enforcing this concept-based way of thinking and labelling. Further, over extensive usage (as the order of submitted sessions approaches dozens or more) I could add a feature to the system that activates previous sessions and/or resources that are relevant to the present session The history effectively becomes a long-term memory from which resources can be activated.

4.4.2 Session-Based Design

The experiment was designed to be structured around discrete sessions. This was intended to simplify the experiment and provide guidance and structure to the users in testing this early iteration of the application.

Interestingly, the structure allows us to glean insight about typical numbers associated with an application of this sort. It was observed that on average there were upwards of 9 or 10 URLs visited during a session. This sounds like a reasonable amount to describe an information seeking session. If the answer isn't found after visiting this many pages, the problem may be too complex for the scope of a single browsing session.

Another interesting number taken from the results is the average of 4-5 tags applied to each session. One user even remarked that after adding that many tags, additional suggestions were no longer relevant as most of the meaning had already been captured. Perhaps this number has something to do with our memory limitations; we can typically story about 4-7 items in short term memory [Card et al., 1986]. After this we would need to enter a higher level of abstraction, perhaps clustering groups of sessions and labeling them.

The session-based design, I believe, was also a contributor to the adaptation of the tag paradigm. Without the session limitation, the experiment kept the same otherwise would be ambiguous in terms of when to add tags, though the nature of the dialogue between human and machine could be tweaked to ameliorate this ambiguity, if it is desired to do so.

Of course the session-based design is disruptive, as reported by several users, as most just wanted the system to start recording as soon as they began a search; without the requirement of pressing the record button or entering a goal. This could certainly be implemented in the future, but internally it would still be good for the system to infer context switches, in order to be able to group resources in related clusters.

4.5 Action Items

Recommendations via a minimal overlay

Several users expressed the notion that the most cumbersome aspect of using the Mental Model Browser application was having to jump back and forth between the feedback tab and the pages of interest. Instead, what they desired was some sort of overlay that took up a small part of the screen space and that would allow them to continue their regular browsing minimally disturbed.

Improving Recommendation Quality

In order to improve recommendations the system needs to be able to obtain a better model of the user's intent. The users that reported seeing interesting recommendations were those that created more complete browsing sessions. They had more User Tags, whether self-chosen or accepted from the recommended tags. This provided a better model for the machine to create queries into the Delicious service.

Automatic Inference of Context Switches

The present application asks users to perform searches within the confines of discrete searches having discrete goals. Further they are asked to explicitly state that goal before starting. It is completely possible, however, to remove this requirement and have the system sense context switches. The way it does this is creates a collective topic models of sub-sequences of browsing and then detects context rifts in the form of a vector distance above some threshold.

Chapter 5

Contributions and Reflections

5.1 Contributions

In my thesis research, I have investigated augmenting human intelligence through computer systems that create representations of the user's working and past knowledge and provide recommendations based on outside data sources. The primary contributions are:

- Design and implementation of the Mental Model Browser web application, a system for enhancing the experience of searching for information on the web through recording a browsing session, constructing a representation through interaction with the user, and providing recommendations.

- Application of a topic model representation to modelling web documents and browsing sessions, and providing recommendations based on predicted structural matches.

- Application of a cross-disciplinary set of background ideas and philosophies in the context of augmenting information-seeking on the web.

- A pilot test of the Mental Model Browser application with 22 users who engaged in a total of 56 browsing applications. The results showed that users readily adapted the workflow of the system and received good feedback and recommendations.

5.2 Future Work

5.2.1 Further Development of the Mental Model Browser

In its present incarnation, the Mental Model Browser enhances the browsing experience through a somewhat rigid dialogue between man and machine. The ultimate goal would be to deploy a derivative of the application into the mainstream of web application usage. In order to achieve this goal, several aspects of the application would have to be altered. Through user testing, I identified that reducing the footprint of the application as much as possible would be key to improving usability. Users found it disruptive to have to jump between tabs, however there were positive reactions to the visual feedback of browsing history, as well as the recommended URLs, which were often resources the user would not have discovered through the standard web search workflow. Another necessary modification to the present application would be eliminating the burden from the user of having to indicate the start and end of a session. The system should track sequences of resources and identify context switches. The system should continue modelling the session and carrying out a dialogue with the user through recommended topic tags, but again, the design would have to be reworked in order to limit disruption.

5.2.2 Capturing other Workflows

The concepts illustrated in this thesis may be extended beyond the domain of web browsing and into other electronic workflows. Analysts, designers, and programmers, among others, would benefit from maintaining a dialogue with a system to track the progress of the present problem solving session and receive feedback and recommendation. Further, the system could enable collaborators to share knowledge, akin to the benefits of EWall. Additionally, such a system may provide a better means of establishing knowledge bases within firms, institutions, or public domain, that trump the present status quo of using wikis.

5.3 Visions for Future Technology

5.3.1 Multi-Modal Activation

Mobile computational devices are prevalent in our daily lives, and their sensing and processing capabilities are rapidly increasing. Soon the avant-garde concept of "life-logging", or recording big chunks of our everyday activities, will become more mainstream. In fact a substantial part of our typical experience can already be recorded through the inputs available on mobile devices - cameras and microphones in particular. The bottleneck is close to shifting from hardware to software that not only organizes data capture, but provides a motivation to do so. A key idea in this thesis is adhering to a goal-centric perspective of our daily tasks. As we begin to collect greater streams of data, we may create software that partitions these streams into distinct tasks and identify salient sensory hooks that can be used as pointers or tags to the data. In the future, the sensors may recognize environment triggers that will activate the information recorded at a previous time and form a connection, quite analogous to the way our minds are already reminded among similar situations. The computational analog, of course, offers a more detailed - if not perfect - recollection of relevant situations. Reality as we know it will be reshaped.

5.3.2 Knowledge-based Social Network

Present social networks are often limited to being extensions of real-world relationships. These networks provide alternate channels of communication and sharing with our friends and acquaintances. However, making new friends through these networks is not a common practice, and at best the relationships may be based on a common interest or hobby, resulting in a weak tie. I propose here, that if we can find a way to divert our information processing behaviors through a system that models and augments these behaviors, the resulting system will naturally evolve into a knowledge-based social network. The benefits of sharing information and having recommendations streaming from colleagues and potential colleagues will outweigh privacy concerns - those that choose to give a little will find they receive much more in return and will be much more enabled than their peers who do not readily subscribe to such a vision. Relationships will more readily form without basis in the real world. The pursuit of knowledge need not inherently be grounded in the real-world as typical relationship building is. I see early adoption of such a system to begin in universities, with graduate students and professors discovering their peers outside of the conference paper cycle. True, attempts at knowledge-based social networks are being made right now (for example academia.edu) but most of these sites are limited to the typical static profile model and do not adequately provide for or incentivize the migration of workflows into the system. Just as email is an indispensable part of our present lives, working and researching in a social environment where machine algorithms assist in sharing information will become the norm.

Appendix A

Experiment Instructions

Instructions

First of all, thank you for helping out with this experiment. The system is not perfect, so please let me know if you come across problems. Email me at glebk@mit.edu.

The main idea of this experiment is to record web browsing sessions where you have a fairly specific goal or question you are trying to answer.

Example Session

Overview

The experiment involves capturing snapshots of problem solving sessions using a web browser, or in other words, capturing **real-time browsing history**. When a user has a question, or goal, he or she may decide to look for the answer online. The user conducts some web searches, visits several web pages, and eventually makes some progress toward answering the question. In this experiment you will be asked to install a browser extension for the Google Chrome Browser. This extension adds a button next to your address bar that lets you start recording a session. The experiment provides a session feedback view at http://singularity.xvm.mit.edu/ that allows you to view and interact with your current session data.

Step 1: Get Google Chrome Extension

This experiment requires Google Chrome and the extension at
http://singularity.xvm.mit.edu/install_extension/

Step 2: Start a Recording Session

Press the button added to your browser (looks like a blue brain). Some red text that says REC should appear over the button to indicate that you are recording. This will open up the live session view page as well.

Step 3: Enter a Session Goal

In the appropriate text input, record a phrase or sentence that describes what question you are trying to answer this session. Feel free to use any of the suggested searches to start:

- How do I short sell stock?
- What programming language is best for web development?
- What are the latest NASA projects?

There's no need to invent random questions to ask. As you do your regular browsing over the next few days, use this extension to record the sessions and provide valuable experimental data. The beauty of the experiment is that it records a behavior that most people already do, which is searching for information on the web.

Please remember to press 'Submit Session' if you are happy with a session and don't mind donating it for analysis.

Step 4: Browse the Web as Usual

Continue browsing until you have accessed several pages and have gained some insight into your question. It is expected that you visit anywhere from 2-10 pages, though more is okay.

NOTE: Extension records urls you visit in all tabs that you open.

Step 5: Add Session Tags

In the Session Tags section* of the feedback page, there is an area for entering tags that describe this session. These tags should typically be single words that describe the session and the kind of resources visited. Imagine if you were trying to find this workspace you created and you remember it by a few meaningful words. Add these to the tags. Feel free to add some tags and then continue browsing. There are also "AI-Recommended Tags" that might be relevant. Add any that you think are good. Some example tags:

- shopping
- programming
- python
- web development

NOTE: If you want to accept an AI-Recommended tag, you need to click on it or type it into the User Tags

Step 6: Delete Irrelevant Resources in Session View

If there are any resources that you feel poorly represent this session or just weren't helpful, go ahead and delete them before submitting the session. You can delete resources and then browse some more as well.

Step 7: Submit Session

Once you've visited several pages, go to the feedback view and press Submit Session. This will stop the browser extension from recording any further.

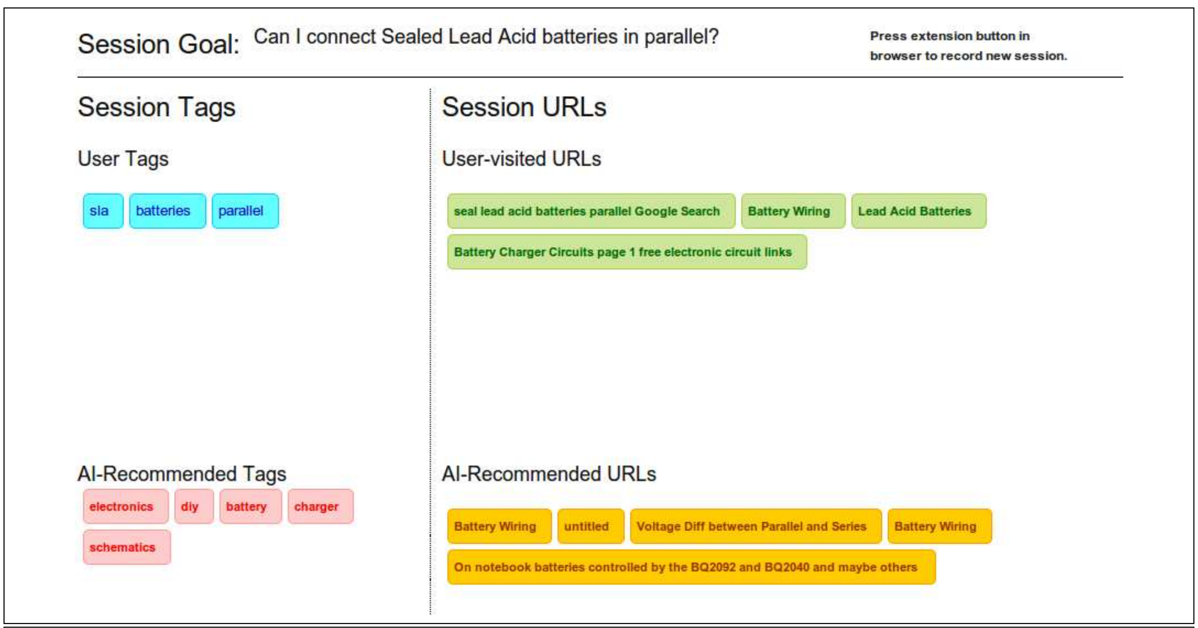
Step 8: Super Brief Survey (after conducting several sessions)

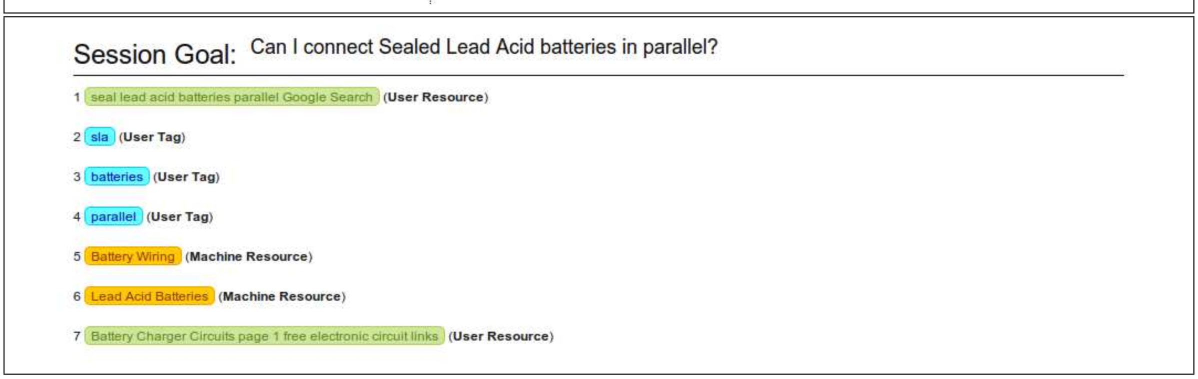
Please fill out the Survey after you've had a chance to submit some searches.

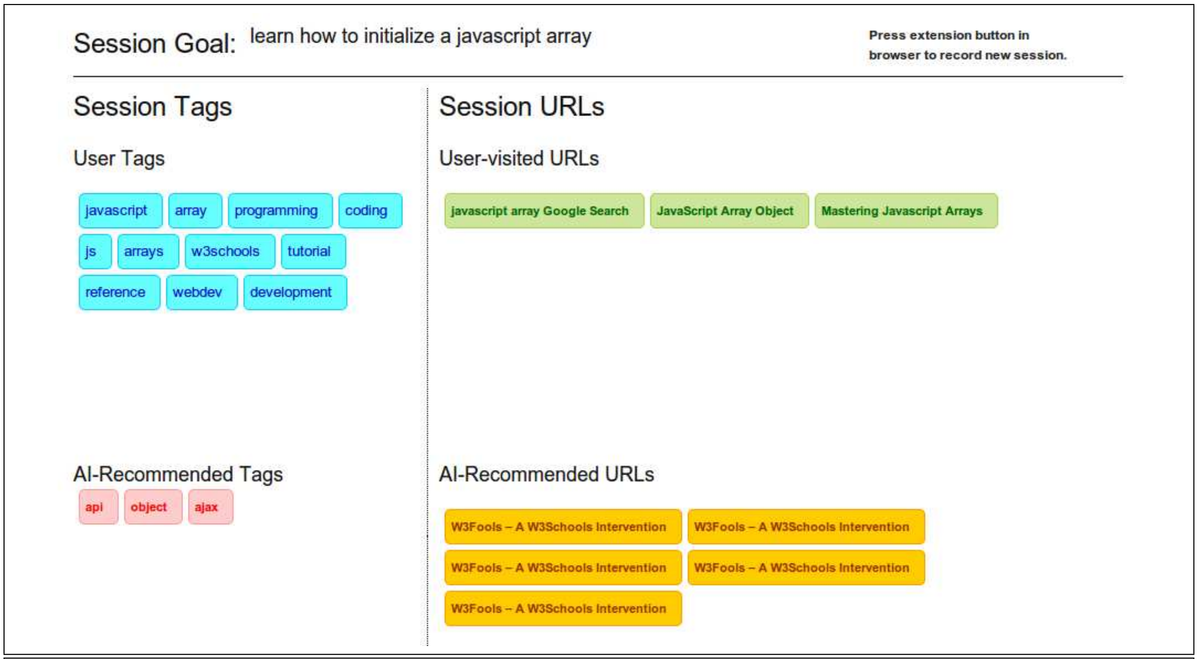
Press extension button to begin recording session!

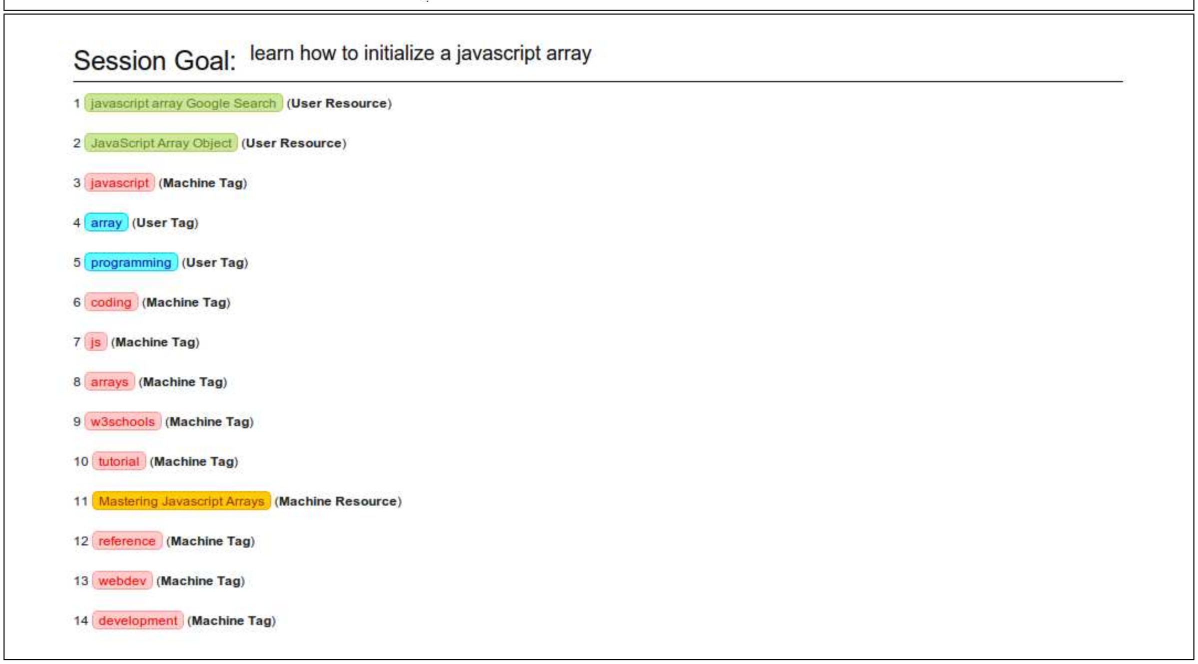
Appendix B

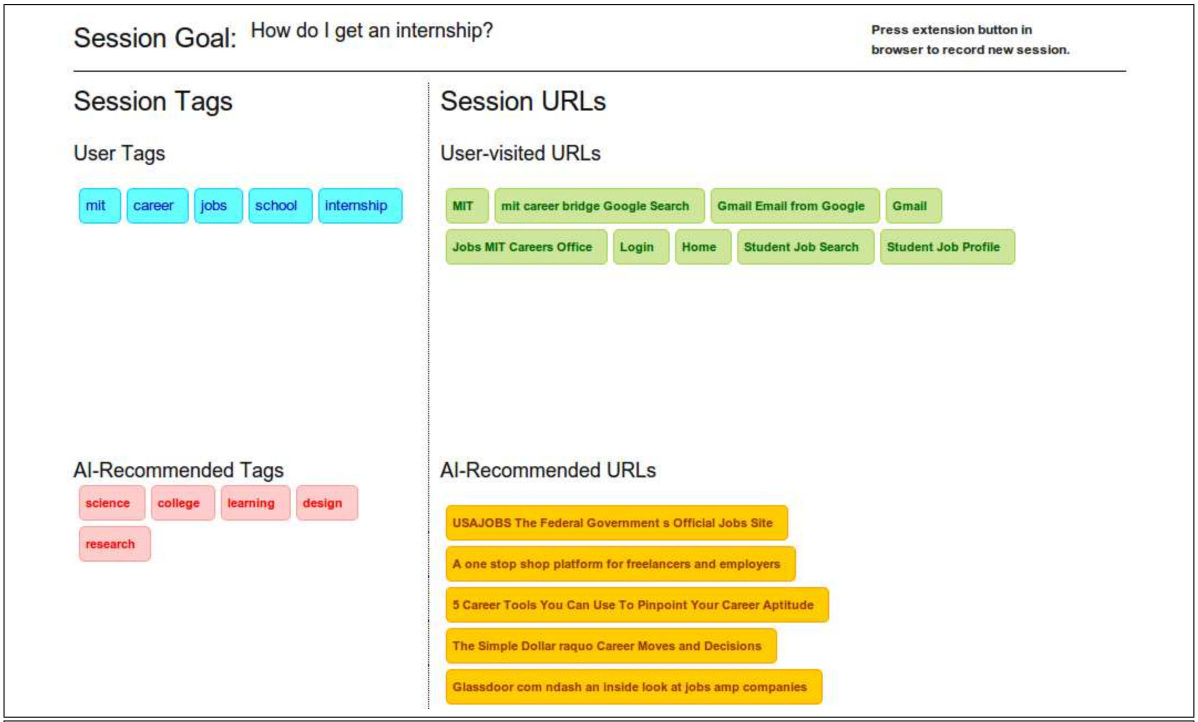
Example User Sessions

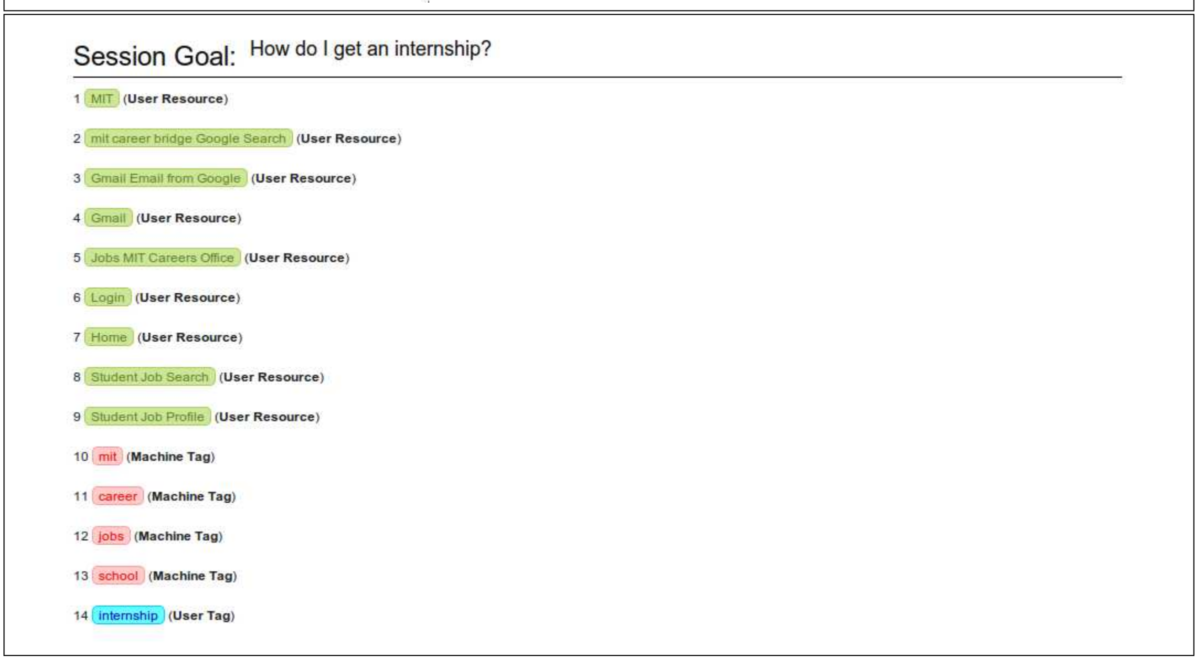





Session Goal: learn how to initialize a javascript array
Press extension button in browser to record new session.
Session Tags
User Tags
javascript
array
programming
coding
js
arrays
w3schools
tutorial
reference
webdev
development
AI-Recommended Tags
api
object
ajax
Session URLs
User-visited URLs
javascript array Google Search
JavaScript Array Object
Mastering Javascript Arrays
AI-Recommended URLs
W3Fools – A W3Schools Intervention
W3Fools – A W3Schools Intervention
W3Fools – A W3Schools Intervention
W3Fools – A W3Schools Intervention
W3Fools – A W3Schools Intervention


Session Goal: learn how to initialize a javascript array
1 javascript array Google Search (User Resource)
2 JavaScript Array Object (User Resource)
3 javascript (Machine Tag)
4 array (User Tag)
5 programming (User Tag)
6 coding (Machine Tag)
7 js (Machine Tag)
8 arrays (Machine Tag)
9 w3schools (Machine Tag)
10 tutorial (Machine Tag)
11 Mastering Javascript Arrays (Machine Resource)
12 reference (Machine Tag)
13 webdev (Machine Tag)
14 development (Machine Tag)


Session Goal: How do I get an internship?
Press extension button in browser to record new session.
Session Tags
User Tags
mit
career
jobs
school
internship
Session URLs
User-visited URLs
MIT
mit career bridge Google Search
Gmail Email from Google
Gmail
Jobs MIT Careers Office
Login
Home
Student Job Search
Student Job Profile
AI-Recommended Tags
science
college
learning
design
research
AI-Recommended URLs
USAJOBS The Federal Government s Official Jobs Site
A one stop shop platform for freelancers and employers
5 Career Tools You Can Use To Pinpoint Your Career Aptitude
The Simple Dollar raquo Career Moves and Decisions
Glassdoor com ndash an inside look at jobs amp companies


Session Goal: How do I get an internship?
1 MIT (User Resource)
2 mit career bridge Google Search (User Resource)
3 Gmail Email from Google (User Resource)
4 Gmail (User Resource)
5 Jobs MIT Careers Office (User Resource)
6 Login (User Resource)
7 Home (User Resource)
8 Student Job Search (User Resource)
9 Student Job Profile (User Resource)
10 mit (Machine Tag)
11 career (Machine Tag)
12 jobs (Machine Tag)
13 school (Machine Tag)
14 internship (User Tag)

Bibliography

Aire. URL http://www.ai.mit.edu/projects/aire.orig/index.shtml.

Delicious. URL http://www.delicious.com.

John Bevilacqua. Decision Making by Agent Committee. Master's thesis, MIT, 2004.

Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1986.

Douglas C. Engelbart. Augmenting human intellect: A conceptual framework. Technical report, Stanford Research Institute, 1962.

M.A. Finlayson and P.H. Winston. Analogical retrieval via intermediate features: The goldilocks hypothesis. 2006.

K Forbus. MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19 (2):141–205, June 1995.

Dedre Gentner and Arthur B. Markman. Structure mapping in analogy and similarity. *American Psychologist*, 52(1):45–56, 1997.

Michael Alan Kahan. Computer Analysis, Learning and Creation of Physical Arrangements of Information. Master's thesis, MIT, 2004.

Paul Erich Keel. *Knowledge Trading*. PhD thesis, MIT, 2004.

David Kirsch. Problem solving and situated cognition. In *The Cambridge Handbook of Situated Cognition*. Cambridge University Press, 2009.

Yao Li. Discovering Structure of Data to Create Multiple Perspective Visualization. Master's thesis, MIT, 2004.

J.C.R. Licklider. Man-computer Symbiosis. *IRE transactions on human factors in electronics*, 1(1), 1960.

M Minsky. A framework for representing knowledge. 1974.

Marvin Minsky. K-lines: a theory of memory. *Cognitive Science*, 4(2), 1980.

Marvin Minsky. *The Society of Mind*. Simon & Schuster, 1988.

Marvin Minsky. *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. Simon & Schuster, 2007.

Allen Newell and Herbert Simon. *Human Problem Solving*. Prentice Hall, 1972.

William M. Pena and Steven A. Parshall. *Problem Seeking: An Architectural Programming Primer*. Wiley, 2001.

Matthew C. Sither. Adaptive consolidation of computational perspectives. Master's thesis, MIT, 2006.

Paul R. Smart, Paula C. Engelbrecht, Dave Braines, Mike Strub, and Cheryl Giammanco. The network-extended mind. In *Network Science for Military Coalition Operations: information Extraction and Interaction*. IGI Global, 2010.