Context-Based Page Unit Recommendation for Web-Based Sensemaking Tasks

Wen-Huang Cheng
Graduate Institute of Networking and Multimedia
National Taiwan University
Taipei 10617, Taiwan, R.O.C.
wisley@cmlab.csie.ntu.edu.tw

David Gotz
IBM T. J. Watson Research Center
19 Skyline Drive
Hawthorne, New York 10532, USA
dgotz@us.ibm.com

ABSTRACT
Sensemaking tasks require that users gather and comprehend information from many sources to answer complex questions. Such tasks are common and include, for example, researching vacation destinations or performing market analysis. In this paper, we present an algorithm and interface which provides context-based page unit recommendation to assist in connection discovery during sensemaking tasks. We exploit the natural note-taking activity common to sensemaking behavior as the basis for a task-specific context model. Our algorithm then dynamically analyzes each web page visited by a user to determine which page units are most relevant to the user's task. We present the details of our recommendation algorithm, describe the user interface, and present the results of a user study which show the effectiveness of our approach.

ACM Classification Keywords
H.4.3 Information Systems: Communications Applications—*Information Browsers*

General Terms
Algorithms, Design, Human Factors

Author Keywords
Recommendation, WWW, Search, Sensemaking

INTRODUCTION
The World Wide Web provides a rich environment for navigating information. Dramatic improvements in search and retrieval technologies have made it extremely easy for people to find a list of web pages most relevant to their interests. For example, finding the web page for a newly released movie is as simple as entering the title into any search engine. However, even for very simple tasks, locating a specific web page is most often just the first step in a much longer process.

For example, consider information seeking tasks where the goal is to locate a specific piece of information which is clearly understood by the user (e.g., the location of a restaurant). First, the user must perform a web search to locate an appropriate web page. Second, after navigating to the identified web page, the user must scan the page's content to understand its logical structure (e.g., *"What information is contained on this restaurant's web page and how is it organized?"*). Third, the person must identify the appropriate section of the web page (e.g., *"I think I'll find the address in the directions section."*). Only then can the user perform the final step in the process: reading the detailed content of the page to locate the specific information of interest (e.g., *"The directions end at 101 Main Street."*).

Search engines assist only with the very first stage of this process. Meanwhile, few tools have been designed to help locate information after arriving at a specific web site. Yet studies of user behavior on the web show that people actually spend just a small fraction of their effort searching for web pages. Instead, users spend roughly 75% of their time in the post-search phase of looking through individual web pages or sites to find the specific content that is the target of their task [17].

The post-search effort to find relevant information within a web page is even more onerous during *sensemaking tasks* where users must make sense out of potentially conflicting information scattered across many locations. Such tasks are typically longer running and more complex than information seeking tasks, requiring users to collect fragments of information from multiple sources and discover connections between them. This process of "connecting the dots" is a difficult challenge in which users must mentally compare the content from a page they are viewing with the set of information found during previous stages of their task. Sensemaking tasks are quite common and include both personal (e.g., researching vacation destinations and choosing where to buy a home) and business (e.g., business intelligence and market analysis) activities.

For example, consider Alice who is relocating to a new city after a promotion at work. She begins to plan her move by searching the web for information (e.g., the quality of schools, neighborhood crime rates, and typical home prices). She notes her findings using a web-based note taking tool such as Google Notebook [8]. Alice continues to explore additional issues (e.g., child care options, health care providers for her husband's rare disease) over dozens of follow-up web sessions spanning several weeks. As her research progresses, she combines information found at various stages into a comprehensive plan for her move. For instance, she vets a web page of potential day care providers against a list of towns in which she previously found apartments for rent. Then, while looking at a hospital's list of specialists for her husband's condition, Alice by chance notices a serendipitous connection. On the same street as the hospital is a day care center and an apartment complex in her price range. With this lucky discovery, Alice decides that this is where she will live and focuses her future research on this key town.

Existing search engines and note taking tools can help users like Alice find specific web pages and save information for future review. However, these are not the most most arduous parts of Alice's task. Rather the bulk of her time is spent reading through each newly visited web page to find any relevant connections it may have with the information in her notes. Due to the volume of information and the length of the task, this can be difficult even when Alice is looking for a specific connection (e.g., finding day care and housing within the same town). It is even more challenging to discover serendipitous connections (such as Alice's discovery of a shared street name) that could go easily overlooked. Yet finding these connections is the essence of sensemaking behavior. Therefore, an effective tool for web-based sensemaking tasks must meet the following requirements:

- **Site Independence:** A sensemaking tool must be independent of any specific site or content provider to allow cross-site connection discovery.
- **Note-Taking Functionality:** A sensemaking tool should allow for the collection of information fragments into a task-specific workspace to help users organize their findings across multiple sessions and sites.
- **Assistance in Connection Discovery:** Most critically, a sensemaking tool should assist the user in performing the difficult process of uncovering connections between their notes and what is currently being explored in their browser.

We know of no existing tool that satisfies these requirements. Therefore, we have developed a smart web-browsing system called the *InsightFinder* which provides traditional note-taking capabilities augmented with a novel technique for context-based page unit recommendation. Our algorithm dynamically analyzes each visited web page to determine the most task-relevant fragments of content. We exploit the natural note-taking activity common to sensemaking behavior as the basis for our task-specific context model. As a result, our algorithm can detect both intended and serendipitous connections between a visited web page and the user's task context.

In this paper, we describe our system's architecture and user interface design, as well review the algorithm used to provide dynamic context-based page unit recommendations. In addition, we present the results from a user study that show that our tool can effectively identify content-relevant page segments in comparison to human judgments of relevance. Moreover, the study findings show that participants using the InsightFinder can identify relevant content within a web page significantly faster than those using a traditional browser.

BACKGROUND

Our work is closely related to several areas of research. These include sensemaking tools, web-based information tasks, and personalization. Here we survey a sampling of this related work.

Sensemaking Tools

Sensemaking has been a focus of HCI research for many years. In the 1990s, researchers recognized the importance of sensemaking behavior in the design of knowledge representational tools, information retrieval systems, and user interfaces [23]. Closely related, information foraging theory [21] was proposed to help understand how users adapt to technology when performing information seeking and consumption tasks. This area of study has recently received renewed attention with a focus on the intelligence analysis domain [11, 22] and the emerging field of visual analytics [29]. As a result, a number of tools have emerged for organizing hypotheses, staging discovered information, and illustrating conclusions. These range from brain storming tools [18] to criminal investigation management [14]. Related efforts in visual analytics have led to systems that closely integrate these techniques with visualization tools [10, 27, 31].

Sensemaking on the Web

The broad availability of information on the web makes it a natural resource for sensemaking activity. As a result, many web-based tools have been developed for this purpose. Most basic are the standard web browser features of bookmarking, the back button, and browsing history. Intelligent approaches can be used to extend these basic tools. For example, Magpie [5] extracts semantics from web content to enable enriched management of browsing history.

In other work, task-centric browser extensions have been developed that let users create per-task collections of discovered information. Some of these (e.g. [15]) provide task-based bookmarks by capturing entire web pages. Others tools (e.g., [1, 8, 26]) support more general note-taking capabilities. While the InsightFinder provides similar note-taking capabilities, it goes further by actively exploiting a user's notes for recommendation. Perhaps most similar is the ScratchPad [9], a tool which uses captured notes to improve re-visitation by recommending previously found information. In contrast, the InsightFinder exploits a user's notes to recommend content from within a newly visited web page.

Personalization and Recommendation

Exploiting user behavior on the web has become an important topic in recent years. For instance, the wisdom of crowds can be exploited in several ways including including social bookmarking [12] and tagging [6]. However, a critical aspect for sensemaking is the unique circumstance of an individual's progress on a specific task, not just the consensus behavior of a crowd. The InsightFinder is therefore more closely related to efforts in personalization, including a large body of work on personalized search that has used implicit (e.g., [3, 28]) or explicit (e.g., [7, 30]) user behavior to improve the set of returned results. However, these tools do nothing to help in the post-search phase which we target in our work. In contrast, ScentTrails [19] does assist in post-search navigation, but it is limited to hyperlinks and requires explicit search terms to entered by the user.

THE INSIGHTFINDER

The *InsightFinder* is a smart web-browsing system that assists in connection discovery during sensemaking tasks by providing context-based page unit recommendations. The InsightFinder allows users to capture information they find while researching a particular topic within a notebook-like sidebar. The captured information is then used to allow a user to re-access their stored information (e.g., as with traditional bookmarks), as well as to facilitate real-time recommendations of relevant page units during the user's future browsing activity.

This section begins with a description of the InsightFinder's architecture and design philosophy. It then describes the system's user interface and the underlying context model used to represent the information captured in a user's notes. Next, it describes the page segmentation algorithm that determines the set of semantic page units from which recommendations should be generated. Finally, it describes the relevance computation algorithm that calculates the relevancy of each page unit with respect to the user's task context.

Overall Architecture and System Design

The InsightFinder has two primary functions. First, like other note taking tools, the InsightFinder allows users to store and organize the notes collected during a sensemaking task. Second, the InsightFinder uniquely exploits this valuable information as an implicit task context to provide page unit recommendations which

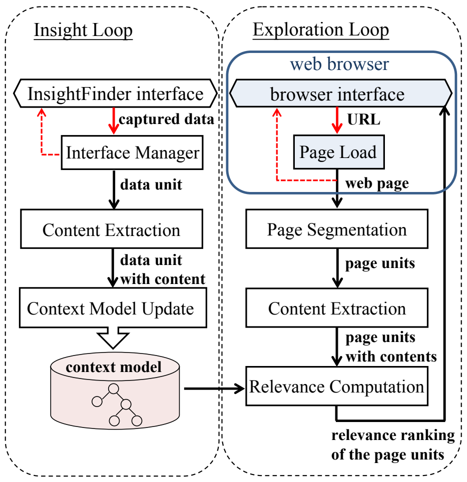


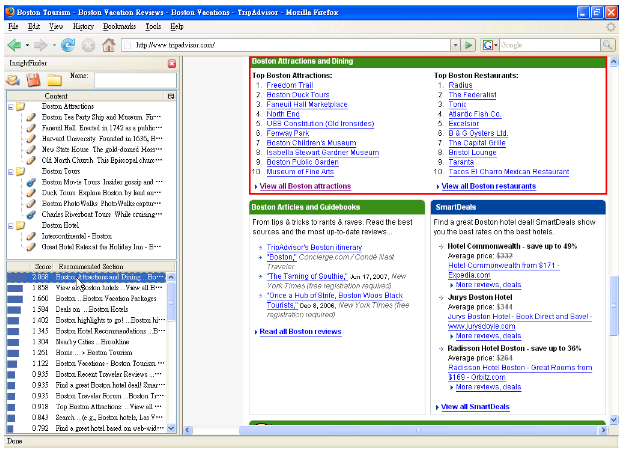
Figure 1. The InsightFinder Architecture. The *insight loop* is executed when users update their notes. The *exploration loop* occurs as users browse the web.

assist during the search for relevant content on newly visited pages.

The InsightFinder architecture consists of two interaction loops as shown in Figure 1. The *insight loop* is triggered directly through the InsightFinder interface, which provides tools for users to record or organize their notes. As the user's notes evolve, the InsightFinder maintains a context model which represents the user's captured data. The *exploration loop* occurs while users interact with the normal browser interface. As users navigate the web, the InsightFinder performs a series of steps each time a new page is visited. At the conclusion of both loops, the InsightFinder provides a ranked list of recommended web page fragments that are most relevant to the content in the user's notes. To provide this functionality, the architecture includes modules for interface management, content extraction, context model management, page segmentation, and relevance computation.

The **Interface Manager** module is responsible for coordinating the real-time interaction process whenever a user enters the insight loop. As a user captures data from the web browser in his/her notes, this component updates the tabular presentation used to display the user's notes within the InsightFinder interface. At the same time, the captured data is forwarded to the content extraction module for further processing. The Interface Manager also monitors for other user behavior (e.g., deleting notes, modifying notes, or reorganizing notes into different folder structures) to ensure that the data stored within the system reflects the displayed set of notes.

The **Content Extraction** module is responsible for extracting content (i.e., primary concepts or ideas) from a *data unit*. Data units are web page fragments, ranging from a single word to an entire page. This module is used in both interaction loops. During the insight loop,


Boston Tourism - Boston Vacation Reviews - Boston Vacations - TripAdvisor - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.tripadvisor.com/
Google
InsightFinder
Name:
Done

Content
Boston Attractions
Boston Tea Party Ship and Museum Fir···
Faneuil Hall Erected in 1742 as a public···
Harvard University Founded in 1636, H···
New State House The gold-domed Mass···
Old North Church This Episcopal churc···
Boston Tours
Boston Movie Tours Insider gossip and ···
Duck Tours Explore Boston by land an···
Boston Photo Walks Photo Walks captur···
Charles Riverboat Tours While cruising···
Boston Hotel
Intercontinental - Boston
Great Hotel Rates at the Holiday Inn - B···

Score	Recommended Section
2.068	Boston Attractions and Dining ...Bo···
1.858	View all Boston hotels ...View all B···
1.660	Boston ...Boston Vacation Packages
1.584	Deals on ...Boston Hotels
1.402	Boston highlights to go! ...Boston hi···
1.345	Boston Hotel Recommendations ...B···
1.304	Nearby Cities ...Brookline
1.261	Home ... > Boston Tourism
1.122	Boston Vacations - Boston Tourism ···
0.935	Boston Recent Traveler Reviews ...···
0.935	Find a great Boston hotel deal! Smar···
0.935	Boston Traveler Forum ...Boston Tr···
0.918	Top Boston Attractions: ...View all ···
0.843	Search ...(e.g., Boston hotels, Las V···
0.792	Find a great hotel based on web-wid···

Boston Attractions and Dining

Top Boston Attractions:	Top Boston Restaurants:
1. Freedom Trail	1. Radius
2. Boston Duck Tours	2. The Federalist
3. Faneuil Hall Marketplace	3. Tonic
4. North End	4. Atlantic Fish Co.
5. USS Constitution (Old Ironsides)	5. Excelsior
6. Fenway Park	6. B & G Oysters Ltd.
7. Boston Children's Museum	7. The Capital Grille
8. Isabella Stewart Gardner Museum	8. Bristol Lounge
9. Boston Public Garden	9. Taranta
10. Museum of Fine Arts	10. Tacos El Charro Mexican Restaurant
View all Boston attractions	View all Boston restaurants

Boston Articles and Guidebooks

From tips & tricks to rants & raves. Read the best sources and the most up-to-date reviews...

- TripAdvisor's Boston itinerary
- "Boston," *Concierge.com / Condé Nast Traveler*
- "The Taming of Southie," Jun 17, 2007, *New York Times (free registration required)*
- "Once a Hub of Strife, Boston Woos Black Tourists," Dec 8, 2006, *New York Times (free registration required)*

Read all Boston reviews

SmartDeals

Find a great Boston hotel deal! SmartDeals show you the best rates on the best hotels.

- **Hotel Commonwealth - save up to 49%**
  Average price: ~~$332~~
  Hotel Commonwealth from $171 - Expedia.com
  More reviews, deals
- **Jurys Boston Hotel**
  Average price: $344
  Jurys Boston Hotel - Book Direct and Save! - www.jurysdoyle.com
  More reviews, deals
- **Radisson Hotel Boston - save up to 36%**
  Average price: ~~$264~~
  Radisson Hotel Boston - Great Rooms from $169 - Orbitz.com
  More reviews, deals

View all SmartDeals

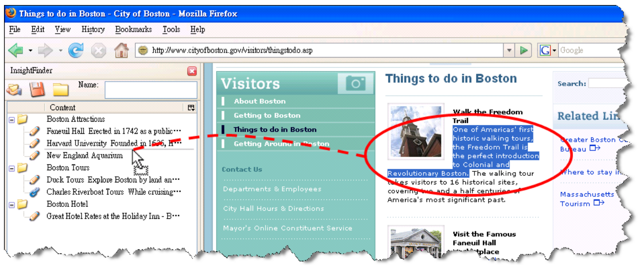
Figure 2. A screenshot of the InsightFinder system.

data units are called *context units* because they correspond to fragments of information from the user's notes. During the exploration loop, data units are called *page units* because they correspond to fragments of a web page as produced by the page segmentation module. The extracted content is represented as a bag-of-words containing a set of meaningful terms selected from the original text contained within the data unit. Currently, a simple unigram algorithm is applied to generate the term set [2]. We clean the textual data by removing stopwords and collect the remaining terms to form the final content term set. We are exploring n-gram term extraction in our future work.

The **Context Model Update** module is the third and final module in the insight loop. It is responsible for integrating the content extracted from the data unit into the context model data structure. The context model, defined formally later in this document, is a graph-based data representation. This module either creates, modifies, or removes nodes within this graph to reflect a user's interaction with his/her notes.

The **Page Segmentation** module is the first module in the exploration loop. Whenever a user navigates to a new web page, this module analyzes the content of the web page in a background process. This module analyzes the structure of the web page and partitions it into a set of smaller fragments called page units. The goal is to isolate individual content units which contain information related to a single topic. The page units are then forwarded to the content extraction module where they undergo the same cleaning process applied during the insight loop. However, unlike the insight loop's context units which are manually created by users, page units are automatically obtained by segmenting every web page visited by a user.

Relevance Computation is the final module in the InsightFinder system. This module uses a relevance metric to compare the page units extracted from the currently displayed web page to the system's context model which represents to the user's recorded notes.


Things to do in Boston - City of Boston - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.cityofboston.gov/visitors/thingstodo.asp
Google
InsightFinder
Name:
Content
Boston Attractions
Faneuil Hall Erected in 1742 as a public···
Harvard University Founded in 1636, H···
New England Aquarium
Boston Tours
Duck Tours Explore Boston by land an···
Charles Riverboat Tours While cruising···
Boston Hotel
Great Hotel Rates at the Holiday Inn - B···
Visitors
About Boston
Getting to Boston
Things to do in Boston
Getting Around in Boston
Contact Us
Departments & Employees
City Hall Hours & Directions
Mayor's Online Constituent Service
Things to do in Boston
Search:
Related Lin
Bureau
Where to stay in
Massachusetts Tourism
Visit the Famous Faneuil Hall

Walk the Freedom Trail

One of Americas' first historic walking tours, the Freedom Trail is the perfect introduction to Colonial and Revolutionary Boston. The walking tour takes visitors to 16 historical sites, covering two and a half centuries of America's most significant past.

Figure 3. Users can record notes by dragging content fragments (links, images, text, or entire pages) from the browser to folders in the InsightFinder.

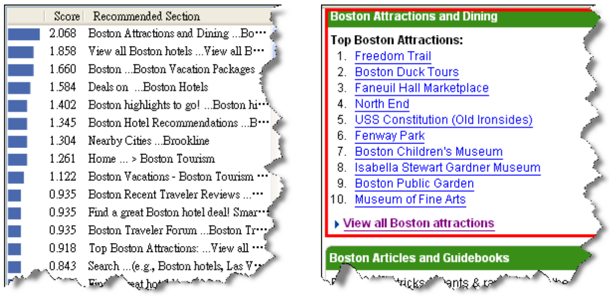
The relevance computation algorithm is part of both interaction loops. In the insight loop, changes to the user's notes trigger a context model update. In response, the system re-evaluates the relevance computation to provide new page unit recommendations based on the new notes. In the exploration loop, the relevance evaluation is performed every time a user visits a new page so that recommendations can be made based on the new content.

The dynamic nature of our relevance ranking algorithm is a critical element of the system's design. Users naturally extend and modify their note-based context models in the process of completing a sensemaking task. In response, the relevance value assigned to a particular page unit will change over time to reflect the evolving context in which the users are working. This design allows users to concentrate directly on their task and record their evolving notes, just as they would with traditional note-taking technologies. Meanwhile, without any additional user input, the InsightFinder automatically and dynamically provides context-relevant page unit recommendations to encourage faster identification of relevant content.

User Interface

The InsightFinder's interface is displayed as a web browser sidebar with two main sections (see Figure 2). The top section of the sidebar contains a note-taking area in which users can record and manage found information across the web. User notes are organized using a tree structure which a user can manage to re-organize and modify entries in his/her notes. New notes or folders can be created directly by the user via the toolbar. Users can also save content from visited web pages by dragging content from the browser as shown in Figure 3. Notes can be created at various granularities, ranging from small HTML fragments to entire web pages. Users can also save individual links, images, or plain text.

Below the note-taking section of the InsightFinder is a tabular view of recommended page units. This section of the tool, shown in Figure 4(a), shows a sorted histogram based on the degree of relevance for each page unit. The most relevant page unit from the user's current web page is displayed at the top of the list, and the



	Score	Recommended Section
	2.068	Boston Attractions and Dining ...Bo···
	1.858	View all Boston hotels ...View all B···
	1.660	Boston ...Boston Vacation Packages
	1.584	Deals on ...Boston Hotels
	1.402	Boston highlights to go! ...Boston hi···
	1.345	Boston Hotel Recommendations ...B···
	1.304	Nearby Cities ...Brookline
	1.261	Home ... > Boston Tourism
	1.122	Boston Vacations - Boston Tourism ···
	0.935	Boston Recent Traveler Reviews ...···
	0.935	Find a great Boston hotel deal! Smar···
	0.935	Boston Traveler Forum ...Boston Tr···
	0.918	Top Boston Attractions: ...View all ···
	0.843	Search ...(e.g., Boston hotels, Las V···

(a)

Boston Attractions and Dining

Top Boston Attractions:

1. Freedom Trail
2. Boston Duck Tours
3. Faneuil Hall Marketplace
4. North End
5. USS Constitution (Old Ironsides)
6. Fenway Park
7. Boston Children's Museum
8. Isabella Stewart Gardner Museum
9. Boston Public Garden
10. Museum of Fine Arts

▸ View all Boston attractions

Boston Articles and Guidebooks

(b)

Figure 4. (a) The InsightFinder presents recommendations as a ranked list with a histogram of relevance. (b) Clicking on a recommendation automatically scrolls the web page and highlights the corresponding content.

Variable	Description
C	A context model
N_C	The set of nodes in C
n_i	An individual node in the set N_C
$\textbf{Type}\{n_i\}$	The type of node n_i
$\textbf{ID}\{n_i\}$	The identifier of node n_i
$\textbf{Payload}\{n_i\}$	The raw data assigned to node n_i
$\textbf{Content}\{n_i\}$	The extracted content assigned to node n_i
E_C	The set of edges in C
e_i	An individual edge in the set E_C
$\textbf{Nodes}\{e_i\}$	The set of nodes connected by edge e_i

Table 1. The context model notation.

the histogram is recomputed with a fresh set of page units each time a user visits a new web page or alters his/her notes. Clicking on an item in the list causes the browser to automatically scroll to the corresponding web page section. The recommended content is them highlighted using a red box as shown in Figure 4(b).

Context Model

The InsightFinder allows users to record information at several levels of granularity and provides tools for the creation, manipulation, and removal of captured notes. Underneath the graphical display of the user's collection of notes is a graph-based data structure, called the *context model*, that mirrors the visual presentation and augments it with additional information required for the page unit recommendation algorithm. This section defines the context model data structure and discusses several ways in which users can interact with the model.

Context Model Definition

A context model C is a disconnected graph defined by a set of nodes, N_C, and a set of edges, E_C. Each node $n_i \in N_C$ corresponds to a data unit explicitly added to the InsightFinder by a user. Similarly, each edge $e_i \in E_C$ corresponds directly to a user-created association between data units. Nodes and edges define the backbone of the context model and all other data is represented as properties of these primary structures.

Context Model Nodes. Nodes represent atomic units of information captured within a user's notes. Each node n_i has several important properties: type, identifier, payload, and content. These properties correspond to either attributes of the information captured by the user, or values computed algorithmically by the system.

The *type* of a node, $\textbf{Type}\{n_i\}$, is initially assigned during the creation of the node and depends upon the type of information being captured. Instead of simply capturing entire pages, the InsightFinder allows users to collect information at a much finer granularity. For example, users can "drag-and-drop" individual links, images, or text fragments from a web page into the InsightFinder. Therefore, whenever a drop event is detected, the InsightFinder analyzes the HTML tags of dropped data to automatically determine the type. For complex fragments, the type field may be multi-valued. For example, dragging an image with links from a web page into the InsightFinder will create a node with dual types: *image* and *link*. Since the type is a native attribute of the captured data, it is immutable unless the stored data itself is explicitly modified by the user.

Nodes also have a unique *identifier*, noted as $\textbf{ID}\{n_i\}$, which includes a time stamp of when the node was created. In addition to providing a unique reference for each node, the identifier can be utilized to build a chronological view of the user's activity. As with to the type field, identifies cannot be modified.

The *payload* of a node, noted as $\textbf{Payload}\{n_i\}$, stores the raw content of the captured information. The specific data placed in this field depends upon the type of node. For example, a node with type *text* will have its $\textbf{Payload}\{n_i\}$ set to a text string, while a node of type *image* will store the actual image data.

The *content* of a node, noted as $\textbf{Content}\{n_i\}$, stores the set of terms extracted by the Content Extraction Module The source of this data depends directly upon the type of node. For example, the content for a node with type *text* is directly extractable from the payload field. In contrast, the content for a node of type *image* is populated from the image's ALT attribute in the corresponding web page if such a value has been assigned.

Context Model Edges. Context model edges are created when users group fragments together in their notes using the folder metaphor provided by the InsightFinder. By grouping nodes in folders, users are implying that there is a common idea or concept shared among the grouped nodes. For example, a user groups a set of nodes within a folder named "Sports" to indicate that all of the captured notes are sports-related.

Within the context model, a folder is represented using a special type of node which is connected by edges to all context nodes placed in the folder by the user. Therefore, an edge is directional and expresses a subsumption relation in which one data unit belongs to a specific folder. The ordered pair of nodes for edge e_i is defined as $\textbf{Nodes}\{e_i\} = \{n_j, n_k\}$ where $n_j, n_k \in N_C$, n_j and n_k are, respectively, a folder and a member node.

Building Context
The context model is used to represent a user's notes as captured during a sensemaking task. The context model is built dynamically by the InsightFinder system while the user performs their normal note-taking behavior. When a user begins a new task, the context model is initialized as an empty graph. The graph then grows as users record information within the InsightFinder interface while browsing the web as shown in Figure 3.

In addition to adding new information, users can also manipulate and organize existing objects within the InsightFinder to reflect changes as their task evolves. Similar to the user's creation of new notes, note modification actions trigger changes within the context model to ensure that it properly reflects all changes to the user's notes. For example, when a specific fragment of data is removed from a user's notes, the corresponding context model node and all associated edges will be deleted.

Other Context Model Interactions
The InsightFinder provides more than basic note taking capabilities. One additional important feature is the ability to maintain unique multi-session context models for each user task. This is especially critical because a complex sensemaking task is often performed over the course of several web browsing sessions. This feature lets users save their work and continue at a later time. Moreover, task-specific context models (e.g., one for "Trip to New York" and another for "Investment Research") allow the InsightFinder to provide task-specific page unit recommendations.

Another feature provided by the InsightFinder is revisitation. Users can easily return to the original page of captured data by simply dragging the data object from the InsightFinder into the main browser panel. Similarly, users can open links saved on the InsightFinder by dragging them to the browser or the URL bar.

Page Segmentation
The InsightFinder is designed to help users quickly locate information within a web page by recommending task-relevant page units. This capability relies on effective segmentation of viewed web pages into representative page units as performed by the Page Segmentation module. The goal is to segment arbitrary web pages into individual units which contain semantically consistent data (i.e., all information in the unit should share a common topic). We employ a structure-based algorithm [13] to perform the segmentation. The algorithm utilizes HTML tag distance to determine page unit boundaries by analyzing a page's DOM tree, a structural representation used internally by most popular web browsers. Specifically, we utilize the TreeWalker DOM object to extract the page's HTML tag hierarchy. In addition to extracting the content for each page unit, we record the pair of DOM indices that correspond to the start and the end positions of the unit within the DOM tree. The indices are used to convey recommendations to the user (via scrolling and highlighting) as shown in in Figure 4(b).

Context-Sensitive Page Unit Recommendation
One of the key components behind the InsightFinder is the *relevance computation* algorithm. As a user browses the web in search of new information related to their task, the InsightFinder employs a relevance algorithm that compares the information stored in the tool's context model with the content of each of the page units extracted from the browser's current page. The page units are ranked based on the computed degree of relevance and then recommended visually to the user.

Relevance computation is an important part of the InsightFinder's ability to support sensemaking tasks. It assists users in "connecting the dots" by notifying the user about potentially relevant connections between their notes and the fragments of information currently on display within their web browser. This feature can prove especially useful in quickly uncovering either intended or serendipitous connections which a user would otherwise overlook or obtain only by tediously analyzing the entire page.

The remainder of this section provides a detailed description of the InsightFinder's relevance computation capabilities. First, it describes a scalar function that calculates the context similarity between context model nodes and page units. Next, building on the similarity function, it defines the relevance function employed by the InsightFinder to estimate how relevant an individual page unit is to a particular context model. Finally, it describes how the relevance function is used to obtain a overall ranking of relevant page units for an entire web page.

Similarity Function
The relevance metric employed within the InsightFinder is built upon a core function which evaluates the similarity between a single node n_i in the context model and a single page unit p_j taken from a web page. This similarity function, noted as $\sigma(n_i, p_j)$, returns a scalar value equal to or greater than zero. A large σ value signals a high degree of similarity, while a value of zero indicates the absence of any relationship. The σ function is defined in Equation 1 where *getContent* extracts key terms from a page unit as performed by the content extraction module, and S is a measure which compares two content units as described later in this section.

$$\sigma(n_i, p_j) = S(\mathbf{Content}\{n_i\}, getContent\{p_j\}) \qquad (1)$$

The content extraction module produces a set of extracted content terms from a data unit (either a page unit or a node in a user's context model). The $S(c_1, c_2)$ metric compares the content of two data units, c_1 and c_2, by examining all possible pairwise combinations of terms $t_1 \in c_1$ and $t_2 \in c_2$ as defined below, where $\hat{S}$ is a pairwise co-occurrence metric.

$$S(c_1, c_2) = \sum_{\forall t_i \in c_1, \forall t_j \in c_2} \hat{S}(t_i, t_j) \quad (2)$$

The current prototype supports two alternative $\hat{S}$ measures: the Jaccard coefficient [25] and the pointwise mutual information measure [4]. These metrics are two of the most popular co-occurrence measures used to evaluate the text similarity between terms in the natural language processing and information retrieval communities [16, 20, 24, 32]. We define both measures below and present experimental results comparing the performance of each technique in the evaluation section of this paper.

Jaccard coefficient. The Jaccard coefficient, noted as $JC(t_1, t_2)$, measures the co-occurrence of two terms, t_1 and t_2, as defined below:

$$JC(t_1, t_2) = \frac{\Theta(t_1 \wedge t_2)}{\Theta(t_1) + \Theta(t_2) - \Theta(t_1 \wedge t_2)} \quad (3)$$

where $\wedge$ denotes the AND operator between terms, and Θ is a function that returns an estimate of the frequency of occurrence of either a single or pair of terms. In our algorithm, we use the frequency of occurrence as measured by the number of web pages which contain a particular term (or pair of terms) according to the Yahoo! search engine. The JC measure is a real function and ranges from zero (no similarity) to one (identity). The value of one indicates that t_1 and t_2 always appear on the same page and never appear in isolation.

Pointwise mutual information. The pointwise mutual information measure, noted as $PMI(t_1, t_2)$, is defined in Equation 4.

$$PMI(t_1, t_2) = \log \frac{\Theta(t_1 \wedge t_2)/M}{[\Theta(t_1)/M][\Theta(t_2)/M]} \quad (4)$$

where M is the total number of pages in the corpus from which the Θ function determines term frequency. We again use the Yahoo! search engine to estimate the Θ and manually estimate M to be sufficiently large. The PMI measure is a real function and its value domain is $[0, \infty]$.

Relevance Function
The InsightFinder's relevance function, $\gamma(C, p_j)$, measures the similarity between an individual page unit p_j and the user's overall context model C. The γ function is built on top of the similarity measure $\sigma(n_i, p_j)$ as defined in Equation 5.

$$\gamma(C, p_j) = \sum_{\forall n_i \in N_C} \omega(n_i)\sigma(n_i, p_j) \quad (5)$$

where N_C is the set of nodes in C and $\omega(n_i)$ is a weight factor for the contribution of node n_i. The γ value is a scalar equal to or greater than zero. Larger values for γ indicate a higher degree of relevance for the page unit p_j.

Conceptually, $\gamma(C, p_j)$ estimates the relevance between C and p_j by summing the similarity between p_j and every node n_i of the context model C weighted by the function ω. In the current implementation we use equal weights where $\omega = 1$ for all nodes. In future work, we plan to examine more complex weighting functions (e.g., based on the centrality of individual nodes) to incorporate the structure of the context model into the relevance calculation.

Relevance Ranking
The InsightFinder uses the relevance metric to recommend task-relevant web page units to the user. Each time a user visits a new page or alters their notes, the relevance function is iteratively called to calculate the relevance of every page unit p_j from the current web page to the current context model C. This produces a list of page units mapped to their affiliated relevance scores. Page units with a score of zero are removed from the list because they correspond to regions of the current web page which have been determined to have no relevance to the user's task. The remaining page units are sorted based on their relevance score and presented to the user within the InsightFinder interface.

EVALUATION
We conducted a pair of user studies to validate the InsightFinder's approach to page unit recommendation. The first study quantified the accuracy of our algorithm's recommendations while the second found that users of the InsightFinder exhibited statistically significant reductions in task completion time. In this section, we describe our prototype implementation and discuss results from both studies.

Prototype Implementation
The InsightFinder prototype is designed as a sidebar-based extension to the Mozilla Firefox web browser. The user interface, shown in Figure 2, is implemented using XUL. The major computational components (e.g., content extraction, page segmentation, relevance computation, etc.) are implemented in Java. JavaScript is used to connect the XUL interface with the Java-based system modules. In addition, the system utilizes a remote web service to gather term frequency data as required by the similarity function.

Ranking Accuracy Evaluation
The page unit recommendation algorithm employed by the InsightFinder examines each visited web page and provides users with a context-based ranking of relevant page units. To evaluate the quality of our system's ranking algorithm, we compared system-generated rankings

	Page 1	Page 2	Page 3	Page 4
User 1	0.3148	0.4236	0.3060	0.4443
User 2	0.5197	0.6517	0.2780	0.4193
User 3	0.2881	0.6024	0.2389	0.2808
User 4	0.3127	0.6433	0.3733	0.3928
User 5	0.3071	0.8143	0.2503	0.5543
User 6	0.3900	0.5726	0.3363	0.1867
User 7	0.5024	0.4305	0.2636	0.3956
User 8	0.6119	1.0000	0.2602	0.4069
User 9	0.3319	0.5143	0.1970	0.3069
User 10	0.4556	0.6107	0.6344	0.5407
Avg.	**0.4034**	**0.6263**	**0.3138**	**0.3928**
Std.	**0.1123**	**0.1737**	**0.1233**	**0.1125**

(a) MHR for Jaccard Coefficient

	Page 1	Page 2	Page 3	Page 4
User 1	0.2741	0.5079	0.3016	0.3038
User 2	0.2733	0.6000	0.3521	0.3556
User 3	0.3269	0.7200	0.3968	0.2149
User 4	0.2928	0.7500	0.4605	0.3169
User 5	0.4044	0.7267	0.2784	0.5460
User 6	0.3022	0.7933	0.2373	0.2416
User 7	0.3467	0.4086	0.4350	0.4036
User 8	0.3756	0.7583	0.3025	0.3645
User 9	0.1789	0.6050	0.3808	0.2538
User 10	0.6444	0.7676	0.8000	0.6143
Avg.	**0.3419**	**0.6637**	**0.3945**	**0.3615**
Std.	**0.1232**	**0.1282**	**0.1591**	**0.1302**

(b) MHR for Pointwise Mutual Information

Table 2. Relevance ranking performance (*MHR*) using (a) Jaccard coefficient and (b) pointwise mutual information similarity measures.

with human judgments of relevance using the same set of page units. We compared the human rankings to both similarity measures supported by our system: the Jaccard coefficient and pointwise mutual information. The methodology and results are described below.

We asked ten human subjects to participate in our experiment. Each participant was told to work on the same sensemaking task: vacation planning for a trip to Boston, MA in the USA. All participants were residents of Taiwan and none had ever visited Boston. Participants were instructed to collect any travel information they considered useful for their trip, such as accommodations, attractions, climate, etc. To avoid bias, participants were not provided with any information about our research goals or the InsightFinder system.

Prior to the study, we manually captured a base set of user notes for the Boston task. These notes were given to participants as an initial planning draft for their task. In addition, we chose four specific travel pages that were each generally related to the city of Boston. Each page was located on the web site of a different content provider.

During their study session, participants were told to visit each of the four test pages one at a time. To avoid bias, participants were not allowed to access the InsightFinder's ranking results. Each user was asked to provide a subjective ranking of the most relevant page units on each page using their own judgment. Participants were not required to rank every page unit. However, we did ask them to provide a list of at least the top five most relevant page units.

We then compared these user generated rankings with the rankings produced by the InsightFinder system when applied to the same four test pages. We scored the comparison between the human and system rankings via a metric we call the *mean hit ratio*, or *MHR*. This metric is similar to mean average precision (MAP) commonly used in the evaluation of ranked retrieval systems, but tailored to the page unit recommendation problem posed on our study. We define *MHR* as a scalar measure of how well two rankings match, ranging from zero to one. A higher *MHR* value indicates a better match between any two given rankings, where the value of one corresponds to a pair of rankings that are identical in every position. *MHR* is defined formally in Equation 6.

$$MHR = \frac{1}{R}\sum_{r=1}^{R} H(r). \tag{6}$$

where r is the participant's rank of a page unit, R is the number of ranked page units, and $H(r)$ is the hit ratio in the InsightFinder's ranking at a given cut-off rank r, capturing the fraction of the top r rankings in common between the rankings. We set $R = 5$ to consider only the top five rankings from each human subject.

Table 2 provides the *MHR* values for each of the 10 users on each of the four pages in our study. The table includes results for both the Jaccard coefficient and pointwise mutual information implementations of the $\hat{S}$ function. Averaged *MHR* values are provided at the bottom of the table.

Discussion. *MHR* captures both precision and recall aspects of our ranking algorithms. The *MHR* values for both $\hat{S}$ metrics average roughly 0.4. This value means that, on average, users are able to find highly relevant page units within the top two or three system generated rankings. Moreover, the standard deviations of *MHR* values are quite small, implying that the InsightFinder's relevance ranking algorithm is stable and relatively consistent across pages.

Comparing the Jaccard coefficient with pointwise mutual information, our results show roughly equal *MHR* values. This finding implies that both metrics perform similarly well in terms of ranking quality. However, in terms of computational complexity, the Jaccard coefficient requires fewer multiplications and executes more quickly. We therefore use it as our preferred metric within the InsightFinder system.

Task Performance Improvement

The overall goal of the InsightFinder's context-based page unit recommendation algorithm is to assist a user

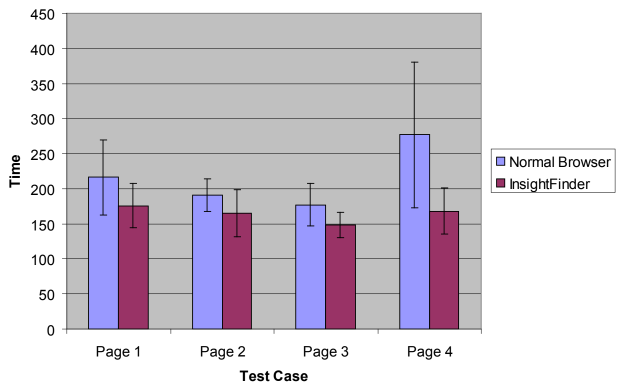


Figure 5. Users of the InsightFinder located relevant information faster ($p < 0.01$) in all four tasks.

in connection discovery while performing a sensemaking task on the web. While the InsightFinder can assist in several ways, we conducted an empirical user study to quantitatively measure one specific benefit of the InsightFinder: the reduction in time required by users to locate relevant information within a web page.

We asked another set of ten participants to complete the same sensemaking task as used in our first study: vacation planning for a trip to Boston. Participants were divided randomly into two groups of five users. As in the first experiment, all participants were given the same initial set of notes to use as a common starting point for the task. The participants in one group were asked to use a normal web browser, while the second group was given access to the InsightFinder's page unit ranking functionality.

Each participant was asked to examine each of the four test pages used in the previous study. For each test page, participants were asked to locate five page fragments that they considered most relevant to their task. For each test page, we recorded the time spent determining the most relevant content. The detailed results of our study are shown in Table 3 and summarized in Figure 5.

Discussion. Our results indicate a strong reduction in time required to complete the study task for participants using the InsightFinder tool. On average, these users completed the task more than 30 seconds faster than their counterparts in the default browser group. This provides statistically significant evidence ($p < 0.01$) that the InsightFinder has measurable reductions in the user time required to identify relevant content within a web page.

In addition, standard deviations of the time measurements are smaller and more consistent within the InsightFinder participant group. We believe this is due to two reasons. First, the InsightFinder's relevance ranking lets users prioritize their efforts by skipping over less useful information more easily. Users can click on

	Page 1	Page 2	Page 3	Page 4
User 11	145	130	176	165
User 12	196	173	128	223
User 13	166	163	149	163
User 14	220	216	153	134
User 15	153	142	136	157
Avg.	**176**	**165**	**148**	**168**
Std.	**31**	**33**	**18**	**33**

(a) Users with the InsightFinder

	Page 1	Page 2	Page 3	Page 4
User 16	171	223	190	440
User 17	187	173	154	167
User 18	221	168	151	242
User 19	193	203	166	305
User 20	306	187	222	229
Avg.	**216**	**191**	**177**	**277**
Std.	**54**	**23**	**30**	**104**

(b) Users without the InsightFinder

Table 3. Statistics of user time spent (in seconds) locating five relevant fragments on a given page both (a) with and (b) without the InsightFinder.

highly ranked page units to jump from one potentially region of the page to another.

Second, the segmentation process itself (as represented by the span of recommended page units) helps users understand the structure of a web page. Even when a user finds the content of a recommended page unit irrelevant, they receive an important benefit. Instead of carefully reading the entire page unit, users can quickly skip over the entire fragment and move on to other more promising regions of the page.

Overall, these evaluation results provide preliminary evidence that the InsightFinder can be a useful tool for sensemaking tasks on the web. While additional technologies and evaluation studies are required, the InsightFinder provides measurable improvements in user performance when compared to traditional web browser interfaces.

CONCLUSION AND FUTURE WORK

This paper introduced the *InsightFinder*, an extension to the basic web browser which provides traditional note-taking capabilities augmented with a novel technique for context-based page unit recommendation. Our algorithm dynamically analyzes each visited web page with respect to a task-specific context model to determine the most relevant page units. We exploit natural note-taking activity, common to sensemaking behavior, as the basis for our task-specific context model.

Results from an empirical user study demonstrate that our tool can effectively identify content-relevant page segments when compared to human judgments of relevance. Moreover, the study findings show that participants using the InsightFinder can identify relevant content within a web page significantly faster than those using a traditional browser.

While our initial approach shows promise, there are a number of topics that could benefit from additional research. In particular, we believe that extending the content extraction module to support n-gram term extraction would provide significant benefit in recommendation accuracy. We also believe that recommendations could be improved by incorporating the structure of a user's notes into the algorithm. We plan to experiment with more sophisticated node weighting functions (ω in our notation) to reach this goal. Finally, we plan to explore richer note-taking interfaces to improve the overall usability of the tool.

REFERENCES

1. Clipmarks. http://www.clipmarks.com/.
2. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Boston, MA, 1999.
3. J. P. Bigham, A. C. Cavender, R. S. Kaminsky, C. M. Prince, and T. S. Robison. Transcendence: Enabling a personal view of the deep web. In *Proc. of IUI*, 2008.
4. I. Dagan, L. Lee, and F. Pereira. Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, 9:123–152, 1995.
5. J. Domingue, M. Dzbor, and E. Motta. Magpie: Supporting browsing and navigation on the semantic web. In *Proc. of IUI*, 2004.
6. M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proc. of Inter. WWW Conf.*, 2006.
7. P. Ferragina and A. Gulli. A personalized search engine based on web-snippet hierarchical clustering. In *Proc. of Inter. WWW Conf.*, 2005.
8. Google Notebook. http://www.google.com/notebook/.
9. D. Gotz. The scratchpad: Sensemaking support for the web. In *Proc. of Inter. WWW Conf. Posters*, 2007.
10. D. Gotz, M. X. Zhou, and V. Aggarwal. Interactive visual synthesis of analytic knowledge. In *IEEE VAST*, Nov 2006.
11. D. Gotz, M. X. Zhou, and Z. Wen. A study of information gathering and result processing in intelligence analysis. In *Proc of IUI Workshop on IUI for Intel Analysis*, 2006.
12. T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools: A general review. In *Digital Library Magazine*, volume 11. Nature Publishing Group, 2005.
13. G. Hattori, K. Hoashi, K. Matsumoto, and F. Sugaya. Robust web page segmentation for mobile terminal using content-distances and page layout information. In *Proc. of Inter. WWW Conf.*, May 2007.
14. i2 Incorporated. Analsyt's Notebook. www.i2inc.com.
15. N. Jhaveri and K.-J. Raiha. The advantages of a cross-session web workspace. In *Proc. of CHI*, 2005.
16. J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of Inter. Conf. on Research on Computational Linguistics*, 1997.
17. A. W. Lazonder, H. J. A. Biemans, and I. G. J. H. Wopereis. Differences between novice and experienced users in searching information on the world wide web. *Jrnl of the American Society for Info. Sci.*, 51:576–581, 2000.
18. Mindjet Corporation. Mind Manager. http://www.mindjet.com.
19. C. Olston and E. H. Chi. Scenttrails: Integrating browsing and searching on the web. *ACM Trans. on CHI*, 10, 2003.
20. T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity - measuring the relatedness of concepts. In *Proc. of the National Conf. of AI*, 2004.
21. P. Pirolli and S. Card. Information foraging. *Psychological Review*, 106:643–675, 1999.
22. P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Inter. Conf. on Intel. Anal.*, 2005.
23. D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card. The cost structure of sensemaking. In *Proc. of CHI*, 1993.
24. G. Salton. *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Publishing Company, 1989.
25. G. Salton and M. J. McGill. *Introductin to Modern Information Retrieval*. McGraw-Hill, 1983.
26. M. C. Schraefel, Y. Zhu, D. Modjeska, D. Wigdor, and S. Zhao. Hunter gatherer: Interaction support for the creation and management of within-web-page collections. In *Proc. of Inter. WWW Conf.*, 2002.
27. Y. B. Shrinivasan and J. J. van Wijk. Supporting the analytical reasoning process in information visualization. In *Proc. of CHI*, 2008.
28. K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proc. of Inter. WWW Conf.*, 2004.
29. J. J. Thomas and K. A. Cook, editors. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Press, 2005.
30. Z. Wen, M. Zhou, and V. Aggarwal. Context-aware, adaptive information retrieval for investigative tasks. In *Proc. of IUI*, 2007.
31. W. Wright, D. Schroh, P. Proulx, A. Skaburskis, and B. Cort. The sandbox for analysis - concepts and methods. In *Proc. of CHI*, 2006.
32. Y. Yang and J. O. Pederson. A comparative study on feature selection in text categorization. In *Proc. of Inter. Conf. on Machine Learning*, 1997.