Deriving Image-Text Document Surrogates to Optimize Cognition

Eunyee Koh
Advanced Technology Labs, Adobe Systems Inc.,
San Jose, CA, 95110-2704, USA
eunyee@adobe.com

Andruid Kerne
Interface Ecology Lab,
Dept. of Computer Science & Engineering
Texas A&M University, College Station, TX 77843, USA
andruid@cse.tamu.edu

ABSTRACT
The representation of information collections needs to be optimized for human cognition. While documents often include rich visual components, collections, including personal collections and those generated by search engines, are typically represented by lists of text-only surrogates. By concurrently invoking complementary components of human cognition, combined image-text surrogates will help people to more effectively see, understand, think about, and remember an information collection. This research develops algorithmic methods that use the structural context of images in HTML documents to associate meaningful text and thus derive combined image-text surrogates. Our algorithm first recognizes which documents consist essentially of informative and multimedia content. Then, the algorithm recognizes the informative sub-trees within each such document, discards advertisements and navigation, and extracts images with contextual descriptions. Experimental results demonstrate the algorithm's efficacy. An implementation of the algorithm is provided in combinFormation, a creativity support tool for collection authoring. The enhanced image-text surrogates enhance the experiences of users finding and collecting information as part of developing new ideas.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Selection process; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia – Navigation.

General Terms
Algorithms, Performance, Design, Experimentation.

Keywords
Surrogates, Information extraction, Search representation.

1. INTRODUCTION
While available media and human needs continue to grow rapidly, people have limited cognitive resources for acquiring information. Humans will benefit from rich sensory multimedia resources found in compound documents as they engage in activities such as research, collecting, and authoring. Liesaputra and Witten showed that users reading HTML books recall the location of a piece of information using images surrounding the text [22]. When there is an illustration following each paragraph in an HTML handbook, readers often used these, in conjunction with the table of contents, to determine which sections were relevant.

The New York Public Library [28] and NSF [27] are among those who have recognized the importance of utilizing image-text surrogates for representing their collections. A surrogate represents an information resource and enables access to that resource [2]. However, these image-text surrogates are carefully prepared by experts, which is expensive. Many collections could benefit from visual representations. In order to alleviate the burden of manually forming such rich representations, we develop algorithmic methods that use the structural context of images meaningful content descriptors in HTML documents to derive image-text surrogates (Figure 1). This work can transform people's everyday experiences with information collections to be more efficient, creative, and enjoyable.

In development of the algorithm, we break the problem down into three stages (Figure 2). We begin with the observation that some pages on the web function as *informative content pages*, which contain materials designed to communicate to users information and knowledge on a particular topic. Others, each of which consists essentially of a set of links, function as *index pages* (Figure 3). Thus, the first stage of the algorithm recognizes whether a page is a content page or an index page. Moving forward, we observe that on the web, even within a content page, we find some areas that function primarily as navigation and advertisement; only part of the document is informative. Thus, only for content pages, stage 2 of the algorithm identifies the most informative content body within a document, and discards the rest. Once this informative content body is identified, we then set out, in stage 3, to extract informative images and the relevant text content. This relevant text content then provides the contextual metadata for the images within it. For example, Figure 1 shows the final output of the algorithm, the image and rich descriptive text extracted from a content page in Figure 3.

Figure 1. Final output of the algorithm, the image and rich descriptive text extracted from a content page in Figure 3 bottom.

DocEng'09, September 16–18, 2009, Munich, Germany.

This paper begins by discussing the background of this research and related work. Then we present our information extraction algorithm, and experiments and analysis to demonstrate the performance of the algorithm. Finally, we discuss the results, draw conclusions and derive directions for future work.

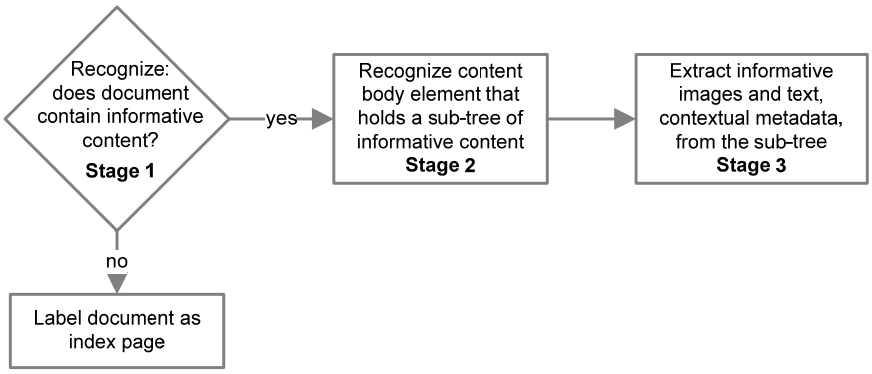

Figure 2. Three stages of our information extraction algorithm. Stage 1 determines the page categorization, stage 2 recognizes the informative sub-tree of the content body page, and stage 3 extracts informative images and text, which are ingredients to form rich document surrogates, from the sub-tree.

2. RELATED WORK

Often, algorithms and technologies are designed and implemented without fully taking into account human cognitive abilities, the ways we perceive and handle information. That is, researchers and engineers often develop computing technologies in relative isolation [15]. Thus, in order to reduce a gap between design, human perception, and technology, we connect findings from cognitive psychology, multimodal surrogate representation, content based image retrieval, and information extraction from web pages. In this section, we discuss prior research methods that support our research and are most relevant to our algorithm. We start by the human working memory that allows people to remain consciously aware of visual and verbal information.

2.1 Cognition of Images and Text

Cognition, according to dual coding theory, involves the activity of two complementary subsystems: a verbal system specialized dealing directly with language and a nonverbal (imagery) system for nonlinguistic objects and events [29]. The internal representations are connected to sensory input and response output systems as well as to each other, so that they can function cooperatively to mediate nonverbal and verbal behavior.

Glenberg *et al.* have established that the combination of an image and descriptive text promotes the formation of mental models, and extends working memory capacity [12]. Carney [5] and Moreno [26] have found that dual coding strategies enhance cognition during educational experiences of digital media. Thus, multimedia combining images and text is a more effective representation than image or text alone.

2.2 Multimodal Surrogate Representations

As the representation of information to the human being for interaction, the surrogate plays an important role in people's finding, discovering and selecting media. A search result snippet is a typical example. Many common surrogates are represented with text-only. Better surrogate representations combine text with other modalities to support cognition.

Marchionini *et al.* investigated the use of multimodal surrogates for video browsing [10][34] by comparing users' performance and experience using different surrogate formats for digital videos. Combined surrogates lead to better comprehension and reduced human processing time. Woodruff *et al.* investigated the efficacy of "enhanced thumbnails" as navigational surrogates for documents [35]. They start with a reduced screen shot of an entire web page. An enhanced thumbnail is annotated with a larger textual "call out," which indicates the presence of a key phrase from a search result set. Users performed significantly better on search tasks with enhanced thumbnails, than they did with text summaries or plain thumbnails. Our research, image-text representation for documents, builds on these results. Instead of the thumbnail images, we use images from the documents and develop an extraction algorithm to reveal significant meanings in a visual form.

2.3 Content-Based Image Retrieval

Research in content-based image retrieval (CBIR) is presented in a comprehensive survey paper by Liu *et al.* [25]. Among all research areas in CBIR, the web image retrieval is the most relevant technique to our approach. Similar to our approach, web image retrieval research extracted text surrounding images from documents to generate semantics for the images [3][11]. However, our approach goes further by expanding from the caption to a more descriptive textual context, at the paragraph level surrounding images. We first recognize informative content pages to identify the pages that contain rich explanatory information. We then extract descriptive text relevant to images. The extracted descriptive text can function as additional contextual metadata for images, which can contribute in enhancing image retrieval results for CBIR research. Further, we utilize extracted informative images and contextual metadata to form surrogates so that people can better understand information.

2.4 Information Extraction Algorithms

Extensive prior research addresses the problem of information extraction from web pages. Detailed background about information extraction research is covered in the survey papers such as [6] and [20]. We focus on methods most relevant to the present research.

Several automated or nearly automated wrapper generation methods have been developed [1][7][8][24][33]. Like some prior researchers [23][31][37], instead of generating wrappers for particular websites, our algorithm is based on generally breaking documents down into block structures, and developing significant features that identify which blocks contain informative content. Like other researchers [7][24][30][31], we start by building a Document Object Model (DOM) [32] tree from each HTML document. Some prior research has defined blocks in the DOM tree by relying only on <table> tags [23], but other tags also can define blocks such as <p> or tags. So we identify content blocks based on the full set of block tags, and then determine

whether a block is informative or not based on features in sub-trees that contain the block. Yi *et al.* also use block tags and ignored the style tags to build the style tree to form a template [36], as did Zhao *et al.*, who developed a search result mining template with block tags [38]. Our previous work also built a Document Surrogate Model from DOM tree only using the block tags, and achieved good performance in web information extraction [18]. The current extraction approach is different from these related works, including our previous work, in that it does not depend on templates and training data, yet it achieves accurate results.

Some research has applied visual features, such as x and y coordinate information rendered in the browser. The VIPS algorithm is based on visual layout [4], and Chen *et al.* [9] and He *et al.* [13] also used the visual features such as the position of the blocks of the pages rendered in the system. A notable aspect of this research is that it depends on using Microsoft Internet Explorer library to perform the layout of web pages, as an intermediate processing stage. A problem with this visual approach is that it requires downloading images and rendering pages to extract features for the algorithm; because this relies on networks downloads, it can be too resource-intensive for use in interactive systems. One end product of this work is the identification of significant visual blocks mostly in index pages. However, researchers need to tune the threshold value for different types of index pages to indentify the fine-grained blocks in the pages. Combining visual features with our approach may be beneficial, but without those features, our research solves the extraction problem with promising performance, and what is better it can operate with fewer resources on diverse platforms.

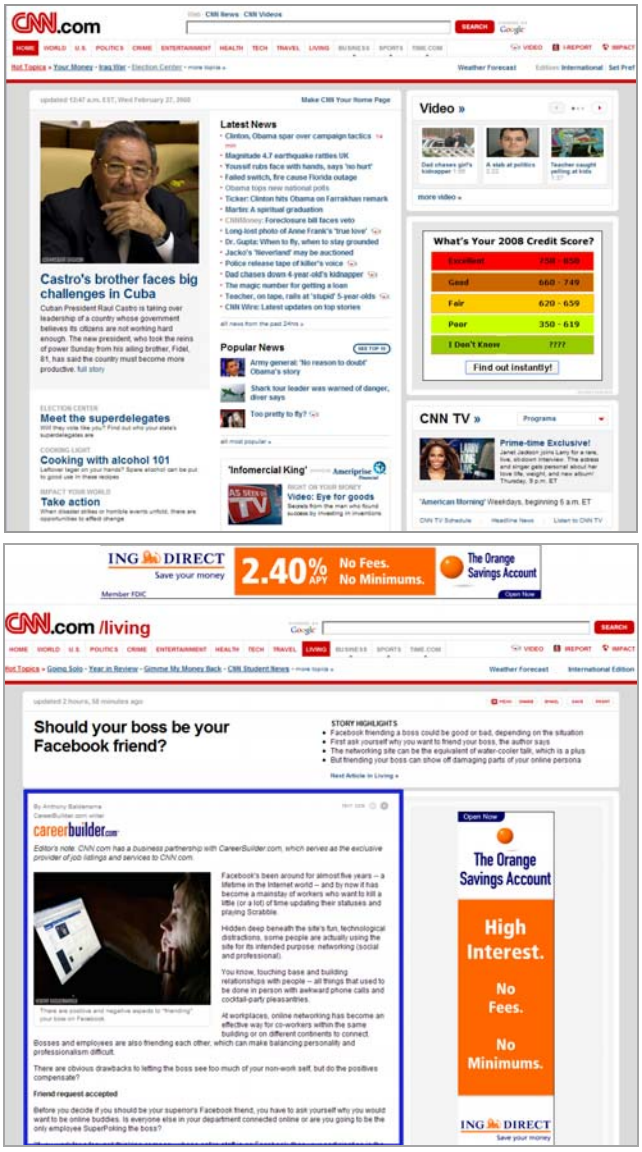

Figure 3. Top: Index Page; Bottom: Content Page.

Recent MSR Asia research has also found that the pictures in web pages could be added into search result pages and provide richer contextual descriptions [21]. In order to generate the image excerpts, they consider the dominance of each picture in each web page and the relevance of the picture to the query. A single "dominant image" is identified for each document. Various image features have been developed to extract the dominant image. Image size was found to be almost twice as good as a detector than any other feature. They claimed that dominant images tend to be photographs containing human faces; this conclusion would seem to be dependent on the type or content in web pages, and not generalizable. Thus, it is important to know the page categorization before we form surrogates. Otherwise, for an example of the Figure 3 top index page, their dominant image detection approach will extract and index Raul Castro's image, which is the top story of CNN only during a certain period of time, and represent Castro's image as an image excerpt for the search result of the query CNN. Instead of seeking a single dominant image, our algorithm identifies multiple image-text surrogates for a single document. This is beneficial for matching with terms from a search query, which may address one particular document sub-component, and more optimally fulfill the user's information needs.

3. INFORMATION EXTRACTION ALGORITHM

The human-centered computing approach of our information extraction algorithm, presented in this section, is based on a perspective on how people comprehend media -- with visual cognition. We would like to provide better, richer representations for informative content pages in collection views, so that people engaged in working with and developing collections can access and understand information more efficiently. Figure 3 shows examples of index and content pages.

Index pages contain collections of information, with a density of hyperlinks. For index pages, collection representations can be changed based on how stylistic choices by the authors create them, but not the actual informative contents that these index pages hyperlink to. Thus, we start by identifying informative content pages for which users need representations. Some informative content pages contain navigation and advertisement parts along with information. Therefore, our algorithm selects the sub-tree that contains the informative contents. Then it extracts the rich media with descriptive text from the sub-tree. With the extracted elements, we can form surrogates for the informative

content pages. The three stages of our algorithm procedure are in Figure 2.

3.1 Categorize Index Page or Content Page

In order to categorize index and content pages, we investigated their primary differences. As our main interest is recognizing the informative content, we examine an informative content body, surrounded by a rectangle in Figure 3 bottom. As only a content page contains a content body, we can categorize the document by determining existence of the content body.

Here are the essential characteristics of the content body compared to the other parts of documents:

- The sub-tree that holds the content body has a greater number of significant words than other parts. The number of significant words is a numerical value that measures the presence of important descriptive and explanatory textual content.
- The ratio of the *stopword* count to the total word count in the content body is minimized in comparison to the other parts. Stopwords are words that are very frequent, and do not carry meaning, such as 'the'. Thus, more significant words consisting of explanatory textual contents are expected in informative content bodies than other parts. We included web *stopwords* such as 'email' or 'advertisement' in the *stopwords* list.
- The content body has more words not surrounded by the <a> tag, as compared to other parts. The words surrounded by the <a> tag mostly represent hyperlinked documents, not the current document. For example, when the hyperlinked words are parts of descriptions in the current document, more information about the hyperlinked text resides in the linked documents, and other significant explanation of information are contained as non-hyperlinked text in the current documents. In the case of the navigation parts of web pages, all texts are surrounded by the <a> tag to represent other pages, so there are no non-hyperlinked words. So, this is also a significant indicator that how many words are not hyperlinked in the content body sub-tree.

We developed DOM node ranking metrics that are designed to assign high weights to DOM nodes with content body characteristics. As the metrics utilize the text in the node, the nodes that do not hold any text will be ignored. Here is the DOM node ranking metric:

$$S_{node} = nw(node.text) - nw(node.atext)$$

$$rank(node) = S_{node} \times [\frac{S_{node}}{nw(node.text)}] \times [\frac{nw(node.text) - nstopw(node.text)}{nw(node.text)}]$$

(1) (2) (3)

nw(text) = word counts of the text
nstopw(text) = stopword counts of the text
node.text = text that the DOM node is holding
node.atext = text that is surrounded by <a> tag and the DOM node is holding

The first parameter among the ranking metrics is (1), S_{node}, the significance of the DOM node. The S_{node} shows how many significant words the node holds for the current page. Sub-expression (2) indicates the ratio of the significant words to the all the words, and sub-expression (3) shows the ratio of non-*stopwords* to all words. These parameters align with the characteristics of content bodies, so the higher the number of significant words, the ratio of the significant words, and the ratio of the non-stop words are then the higher the rank of the node will be and the probability of the node to belong in content body will become higher.

We created a data structure that maintains the highest rank nodes in sorted order. While parsing a page and building the DOM tree, the algorithm discovers DOM nodes and calculates ranking weights (see Figure 4). The data structure is filled by these nodes with rank greater than 0. The nodes for which the rank is 0 contain text that is surrounded by links or *stopwords*. This sorted data structure is iteratively updated. When parsing is finished, the data structure will contain the highest rank nodes present in the DOM tree.

Then, the algorithm iterates through the data structure to find a common ancestor node that holds these highest ranked nodes. If the algorithm cannot find a common ancestor node for any of these highest rank nodes, it recognizes this page as an index page. Otherwise, the algorithm recognizes the page as a content page, and the common ancestor node is identified as the content body node.

Instead of breaking down the document from the root DOM node, we took a bottom-up approach from the highest ranked nodes to determine the content body. If we took a top-down approach, we would need to calculate and compare the ranks among different combinations of sub-trees in the DOM to find the fine-grain content body. The bottom-up approach reduces the operation's computational complexity by maintaining the sorted data structure for the highest ranked nodes. Thus, the algorithm determines the categorization of the page while it is parsing the HTML page and building the DOM tree.

Below is the algorithm for the categorization of an index page or a content page.

Algorithm: Categorization of an index page or a content page

Require: an HTML page

Ensure: page category (either an index page or a content page)

1. **SortedArrayList** highRankNodes[k]; (sorted data structure; k=10 in the current implementation)
2. **Node** contentBody;
3. **while** building a DOM **do**
4. Maintain the highRankNodes;
5. **end while**
6. **while** iterate highRankNodes **do**
7. Check the common ancestor node of the each entry;
8. **end while**
9. contentBody = the common ancestor node that holds most of the nodes in highRankNodes;
10. **if** contentBody.exist() **then**
11. **return** "content page";

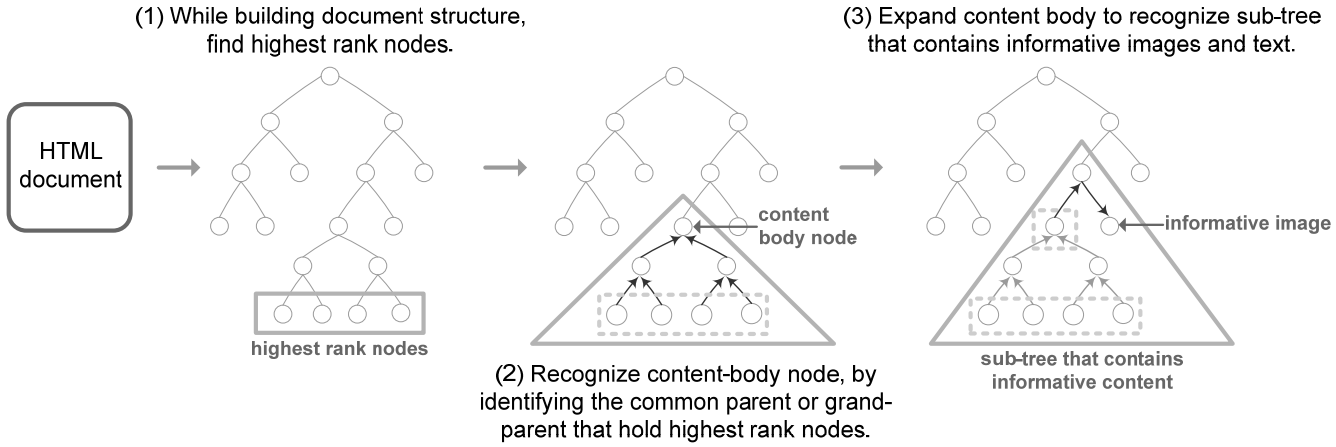

Figure 4. The algorithm recognizes the content body node by identifying the highest rank nodes in the DOM.

12. **else**
13. **return** "index page";
14. **end if**

3.2 Recognize Informative Content

The content body node determined by the previous algorithm holds the informative text, but it does not necessarily hold the informative images. The common characteristic is that the informative images reside in the same branch from the root DOM node and are closest to the informative text. The informative images reside in the context of where the informative text is, and their closeness in the document structure shows meaningful relationships between images and text.

Thus, we determine the parent of the content body node as the sub-tree that holds informative content. The sub-tree is presented in Figure 4, (3) with the triangle region.

3.3 Extract Informative Images and Text

After we determine the sub-tree that holds informative content, we identify the informative images and contextual text within it. Even in the informative sub-tree, there could be non-informative images such as copyright or icon images. Thus, in the selection process, we utilize these features of informative images:

- Informative images reside in the sub-tree of the nodes that hold informative content.
- Image size: Small images are usually icons or copyright images. We reject images that are too small (width is lower than 24 pixels and height is lower than 35 pixels).
- Image aspect ratio: Navigation bars or advertisement images mostly have a high aspect ratio. We reject images with a high aspect ratio (larger than 0.9).
- Text in the image URL or `alt` attribute: Eliminate images that have web stop words like 'ad', or 'advertisement' words.

The threshold values for the image size and image ratio are derived from experience in development and use of the visual collection representation system, combinFormation, with web pages for more than six years (Section 5 and [16]). The threshold values have been utilized and tuned as part of user studies. We derive our algorithmic methods by employing continuous iterative design and experimentation.

In each image node, the size of the images can be specified in attributes of the HTML, but it is not required for document authors to provide this information. We use the attribute values when they are specified, but when they are not, we need to download the images in the content body to determine their sizes. We tried to use the presence of images' hyperlinks as a feature, but while in most cases these refer to other documents, in some sites they are used to show a bigger size of the informative image. Thus, we didn't use it as a feature to recognize the informative images. Still, in most cases, images that have hyperlinks are highly probable to represent the hyperlinked document, not the current one. Future research will use the mime type of the hyperlink destination to incorporate this feature.

Associated text for an image is clipped from the informative text that resides in the sub-tree of the determined content body node. We extracted available caption text by finding the nearest text to each informative image in the DOM tree. If the caption text or alt text is available, we select an associated text context from the content body, using a co-occurrence analysis of terms, in order to expand the representation of the image's meaning. Using these methods, we derive combined image-text surrogates for each content page document. While the image-text surrogates serve as cognitively rich representations of the document, the associated text context also functions as a rich form of metadata for each image.

4. EXPERIMENTS

In this section, we report empirical results obtained by applying our information extraction algorithm to determine the categorization of web pages, identify the content body from the content pages, and recognize the informative images and text within the informative sub-tree in the context of the content body node. As there were no publicly available test datasets that are appropriate for our experiments, we collected our own test dataset from the Web to demonstrate the efficacy of our algorithm.

4.1 Datasets

We required appropriate test data to validate the algorithm. We could not use the TREC test dataset because we required that the

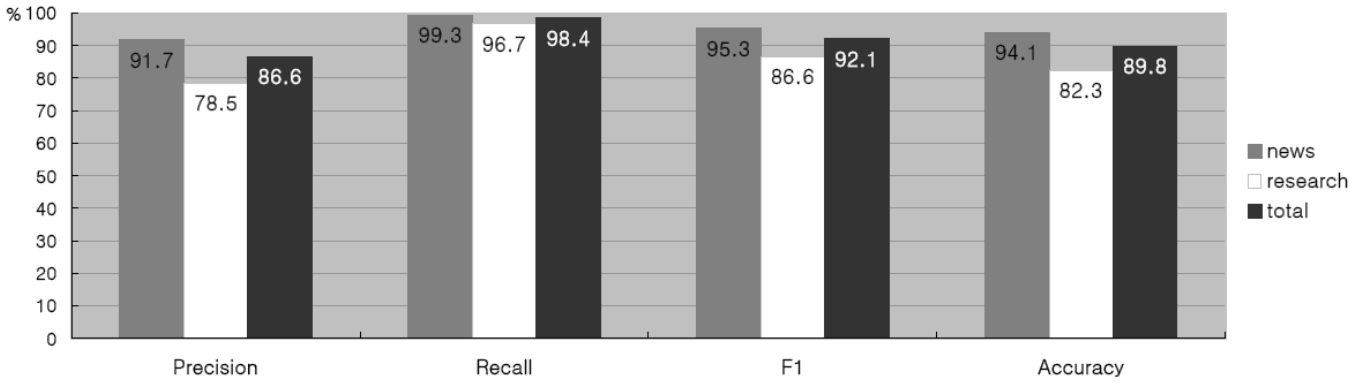

Figure 5. Experiment results show Precision, Recall, F1, and Accuracy for the news collection, research collection, and total collection of news and research.

informative images and text in each document be labeled. We couldn't use the other researchers' datasets which are publicly available (e.g., OMINI: http://disl.cc.gatech.edu/Omini/), because they are solving a different problem, data record extraction, so their test datasets mostly consist of search result pages. There are some researches that have solved similar information extraction problems from news web pages [30][31] and they used labeled news pages like us. However, they have not made their test datasets publicly available.

So, we built a digital library system for labeling and managing the test collection, and an appropriate test dataset [17]. So far, we have collected and labeled 239 content pages from news sites, which are 80 pages from CNN, 52 from the BBC, 54 from ABC, and 53 from Scientific American. We also collected index pages from same sites. They are 27 pages from CNN, 77 pages from BBC, 23 from ABC, 36 from NYTimes, and 15 pages from Scientific American. We call this test dataset the *news collection*. We created another test dataset, the *research collection*. In this collection, we collected pages from university labs and research center sites such NSF, PARC, Microsoft Research, IBM Research, and Los Alamos Laboratory. We collected 151 research content pages and 103 research index pages. This is on-going research. We will continue to collect and label more test pages for our research community. The test dataset is publicly available to researchers at

http://ecologylab.net/testcollections/

4.2 Evaluation Metrics

In a statistical classification task, the *precision* for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the class) divided by the total number of elements labeled as belonging to the class (the sum of true positives and false positives). *Recall* is defined as the number of true positives divided by the total number of elements that actually belong to the class (i.e. the sum of true positives and false negatives). F-measure is the weighted harmonic mean of precision and recall. The traditional F-measure, known as *F1*, is the evenly weighted mean of precision and recall. The *accuracy* is the number of all correctly labeled among the total elements.

4.3 Experimental Results

We present the performance result of the three parts of our algorithm separately, and discuss the overall results. The results show that our algorithm achieves high accuracy in finding informative images and associated text from documents by leveraging the DOM structure and semantic features associated with images and text.

4.3.1 Page Categorization

In the news collection, we had 277 content pages, and 177 index pages. The algorithm recognized 275 pages correctly as content pages, while miscategorizing 2 pages. It also recognized 152 pages correctly as index pages, and failed to categorize the remaining 25 pages. The precision is 0.917, recall is 0.993, and the F1 is 0.953. The results demonstrate the effectiveness of the algorithm (see Figure 5).

In the research collection, we had 151 content pages, and 103 index pages. The algorithm determined 146 pages correctly as content page and missed 5 pages. 63 index pages are correctly categorized as index and 40 index pages are categorized incorrectly. Thus, the precision of the algorithm with the research collection is 0.785, recall is 0.967, and F1 is 0.866 (see Figure 5).

We integrated the news collection and research collection to analyze the performance of the total collection. With the total collection, the prevision is 0.866, the recall is 0.984, and the F1 is 0.921 (see Figure 5).

We investigated the reason why the precision is lower than the recall by investigating failed index pages in categorization. The reason is that the problem index pages contain not only links but also substantial informative content. One example of these pages is in Figure 6. Small gray rectangle boxes in Figure 6 are highlighting the informative content text, so the HTML nodes holding the text will be ranked high. The parent node of those nodes is holding all the highest rank nodes, so the algorithm will determine the parent node as a content body. As this page has a content body, the categorization of the algorithm will be identified as a content page.

4.3.2 Informative Content Body Detection

In the news collections, 237 pages are labeled among the 277 content pages for use in our algorithm evaluation. The label contains where the content body is and what are the informative

images and text. Among the 237 labeled content pages, the content body nodes of 203 pages are correctly determined. The accuracy of determining the content body is 0.857, and the error is 0.143. Among the research test collection, 126 pages out of 146 labeled content pages were correctly identified. The accuracy of determining the content body is 0.863, and the error is 0.137.

We investigated pages that have failed in the content body detection. The algorithm failed because the content body nodes detected by our algorithm were not exactly the labeled nodes. However, the detected nodes were still holding the content body. The example the failure page in Figure 7 shows why we failed to determine the labeled content body. Even for a human, it is difficult to identify the content body node for this example. The labeled outer rectangle border shows what is labeled and it includes the navigation parts in the right side. The inner rectangle border shows what the algorithm determined as a content body, and it excludes the title part of the article. However, there is no node that is holding both title and the article content in this example page.

From this analysis, we find that our algorithm performs accurately in most cases, but that sometimes it is difficult to judge which node to identify as a content body node that holds informative elements in its sub-tree. For the failed cases in the content body detection, as the recognized nodes hold the informative content, they are not really failed cases.

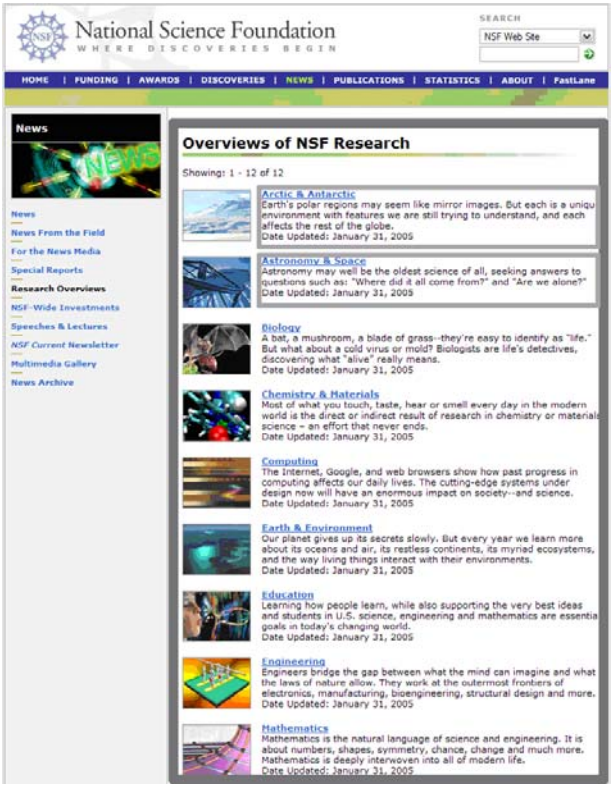

Figure 6. An index page from NSF web site. Even though this is an index page, unlike other index pages, it has informative images and descriptive text information like in a content body. Thus, our algorithm fails to determine these pages as index pages.

4.3.3 Informative Image Detection

Our algorithm correctly detected 222 images out of 237 pages that are labeled from the news collection, and 128 images out of 146 pages. All of the pages from which the algorithm did not extract informative images are text-only content pages, so there are no labels for informative images as well. Therefore, we could determine all of the informative images from the labeled informative content pages.

The performance of the image detection algorithm is better than the content body detection algorithm because the error in the content body detection is not really misidentifying the content body as explained in the previous section. The performance of informative image detection demonstrated that the informative images all reside in the sub-tree of the content body detected from the routine of the algorithm in the previous section. This means that from the correctly recognized content pages, we can extract images and contextual text accurately.

As both the informative images and text reside in the sub-tree of the informative content, in the same document context, we can associate the image with the text from the content body node.

5. CASE STUDY: combinFormation

combinFormation is a collection authoring system that integrates searching, browsing, and exploring information on the Web [14][16][19]. The system provides a visual cognitive interface, using interactive temporal media elements generated by

Figure 7. An example page that failed in determining the content body node. The outer rectangle border shows what is labeled as the content body block, and the inner rectangle border shows what the algorithm determined as the content body.

a computational semantic modeling structure with the human-in-the-loop algorithm. People can initiate the system by specifying search queries or clicking one of the interesting collections of documents provided by the system. Then, the system generates relevant document surrogates as annotated visual media elements over time. People can interact with them by expressing interest or manipulating them based on their own understanding by using the design tools. The system agents respond to a person's interaction by generating more relevant media. With combinFormation, the system and participants can work together to discover serendipitous media and author new media compositions which can be shared with others and published on the Web. Our research has established that this mixed-initiative system supports people in creating and developing new ideas while interacting with found visual information [16].

We have implemented the presented algorithm in combinFormation to derive image-text surrogates, a better visual and semantic representation. As the algorithm is executed during document parsing, there is no bottleneck in people's interaction with the system. We observe significant improvement in combinFormation's performance with the new algorithm, in the form of increased relevance of surrogates, and higher quality metadata. For each image surrogate we included and additional metadata field, 'context', to present our rich descriptive text extracted from the document context through our algorithm (see Figure 8).

In future work, we expect to conduct user studies to demonstrate that this enhanced combinFormation will better support people in information discovery and authoring new creative media collections. By *information discovery*, we mean tasks that involve problem formation and having ideas while searching for, collecting, and organizing media [16].

6. CONCLUSION

The performance of the algorithm is promising. We found over 90% recall with all of the news collection, research collection, and total test collection. The average precision was 85%, which is almost as good. The reduction in the precision was because the algorithm fails to categorize index pages that contain substantial information about their hyperlinks. The algorithm's struggles with these cases are not surprising, as they are also difficult for human beings to label. Thus, future research will work on further

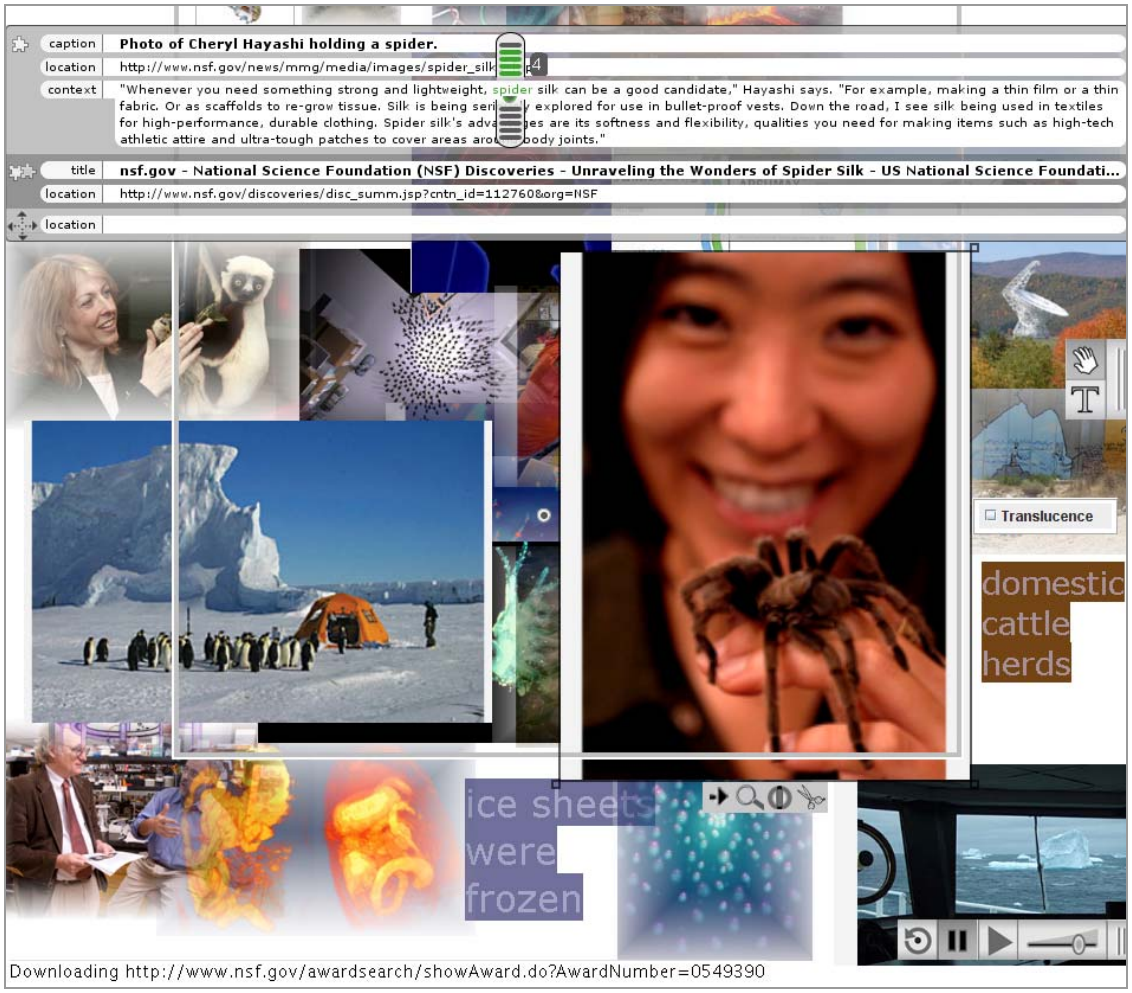

Figure 8. Semantic metadata details-on-demand of an image surrogate with rich contextual metadata in the 'context' field, recognized by the current algorithm, in a composition space created with the collection authoring tool combinFormation.

defining the semantics of these hybrid "index content" pages like a page in Figure 6. We will also continue to collect and label more and various types of test pages to validate and improve our algorithm, and share the test data collections with the research community.

We will refine our algorithm and metrics to determine these index content pages by ranking the content body node with the link threshold. Or, we can recognize recurring similar sub-tree patterns, which are inside the content body node of the index content pages. Then, we will work on recognizing their entry components and extracting useful semantics from them.

With the content pages recognized by the algorithm, the next stages of the algorithm can extract the informative images and descriptive text accurately. We demonstrated that determining the informative sub-tree from the content pages is a significant step for the accurate information extraction.

7. DISCUSSION

The presented algorithm recognizes significant descriptions for multimedia content, specifically images, from document context. The algorithm can easily be extended beyond images to recognize other media embedded into web pages, such as video, Flash, SVG, or audio. The approach will be similar. For example with a YouTube reference, the revised algorithm will recognize content pages that contain video elements. The algorithm will locate each video reference in the content body of the page, connecting each video in the sub-tree of the content body with descriptive text, which can function as metadata that describes it. With this extracted media and associated text in the document structure context, we can form visual multimedia surrogates for the documents.

The algorithm can enhance search engines, by recognizing different document categorizations such as text-only documents or documents full of images with fewer descriptions. When Stage 2 of the algorithm recognizes informative content, cases will arise in which Stage 3 cannot recognize visual media elements. This means the document contains only text information, or all the images in the document are not informative, such as advertisements. When the algorithm cannot recognize informative content in a document, it means that the document itself does not contain coherent information on a focused topic. Or, based on number of informative images inside such documents, those documents may contain only images with fewer descriptions like product search list pages at Amazon. Thus, the quality of the informative content can be further incorporated into finding document categories and the categorization information can help precisely ranking result documents for certain queries in search engines.

Fundamentally, search engines can use the algorithm to derive image text surrogates to present result sets that optimize cognition for users. First, the search engine would use its normal methods to identify result documents for a query. Next, the method of the previous paragraph would be applied to adjust rankings to favor content pages. Then, for each result document, the presented algorithm would be run, deriving a set of image-text surrogates. The one with the best match to the query can then be used to select the surrogate to represent each document.

We have taken a human-centered computing approach to collection representation by identifying the representations for documents that will best support cognition, and have developed an algorithm to automatically derive these image-text surrogate representations. Our algorithm will support automatically generating cognitively and aesthetically better representations that improve the user experience: image-text surrogates. This is of value for digital libraries, which are now embracing not only their own contents but also supporting documents from the Web. Further, our algorithm will recognize the document category and the existence or absence of significant informative contents within those documents. This method will help digital libraries, search engines, collection visualization tools, and other information systems to automatically determine whether the documents are worth including in their collection repositories, and how to rank them for the user.

8. ACKNOWLEDGMENTS

Support is provided by NSF grants IIS-0633906 and IIS-0747428. We would like to thank Laurie Byrum, Adobe Systems and anonymous reviewers for useful comments to improve this paper.

9. REFERENCES

[1] Arasu, A., Garcia-Molina, H., Extracting Structured Data from Web Pages, *Proceedings of SIGMOD 2003*, 337-348.

[2] Burke, M., Organization of Multimedia Resources, Hampshire, UK: Gower, 1999.

[3] Cai, D., He, X., Li, Z., Ma, W-Y., Wen, J-R., Hierarchical Clustering of WWW Image Search Results using Visual, Textual and Link Information, *Proc Multimedia 2004*, pp. 952-959.

[4] Cai, D., Yu, S., Wen, J-R. & Ma, W-Y., VIPS: a Vision-based Page Segmentation Algorithm, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[5] Carney, R.,N., Levin, J.R., Pictorial Illustrations Still Improve Students' Learning From Text, Educational Psychology Review 14:1, March 2002.

[6] Chang, C., Kayed, M., Girgis, M. R., Shaalan, K. F., A Survey of Web Information Extraction Systems. IEEE TKDE 18(10), 2006.

[7] Chang, C., Lui, S., IEPAD: Information Extraction Based on Pattern Discovery, *Proc WWW 2001*, 681-688.

[8] Crescenzi, V., Mecca, G., & Merialdo, P. RoadRunner: Towards Automatic Data Extraction from Large Web Sites, *Proc VLDB Conference, Very Large Data Bases 2001*, 109-118.

[9] Chen, J., Zhou, B., Shi, J., Zhang, H., Fengwu, Q., Function-based object model towards website adaptation, *Proc WWW 2001*, 587-596.

[10] Ding, W., Marchionini, G., Soergel, D., Multimodal Surrogates for Video Browsing, *Proc JCDL 1999*, 85-93.

[11] Feng, H., Shi, R., Chua, T-S., A bootstrapping framework for annotating and retrieving WWW images, *Proc Multimedia 2004*, pp. 960-967.

[12] Glenberg, A.M., Langston, W.E., Comprehension of illustrated text: Pictures help to build mental models, Journal of Memory & Language, 31(2):129-151, April 1992.

[13] He, X., Cai, D., Wen, J-R, Ma, W-Y, Zhang, H-J, Clustering and searching WWW images using link and page layout analysis, *ACM TOMCCAP*, 3(2):Article No. 10, May 2007.

[14] Interface Ecology Lab, combinFormation, http://ecologylab.cs.tamu.edu/combinFormation, last visited 01/28/2009.

[15] Jaimes, A., Gatica-Perez, D., Sebe, N., Huang, T.S., Human-Centered Computing: Toward a Human Revolution, IEEE Computer 40(5), 2007, pp. 30-34.

[16] Kerne, A., Koh, E., Smith, S.M., Webb, A., Dworaczyk, B., combinFormation: Mixed-Initiative Composition of Image and Text Surrogates Promotes Information Discovery, *ACM TOIS*, 27(1) Article No. 5, December 2008.

[17] Koh, E., Kerne, A., Berry, S., Test Collection Management and Labeling System, *Proc ACM Document Engineering 2009*.

[18] Koh, E., Caruso, D., Kerne, A., Gutierrez-Osuna, R., Elimination of Junk Document Surrogate Candidates through Pattern Recognition, *Proc ACM Document Engineering 2007*, 187-195.

[19] Koh, E., Kerne, A., Webb, A., Damaraju, S., Sturdivant, D., Generating Views of the Buzz: Browsing Popular Media and Authoring using Mixed-Initiative Composition, *Proc Multimedia 2007*, 228-237.

[20] Laender, A., Ribeiro-Neto, B., da Silva, A., Teixeira, J., A Brief Survey of Web Data Extraction Tools. *ACM SIGMOD Record*, 31(2), 2002.

[21] Li, Z., Shi, S., Zhang, L., Improving Relevance Judgment of Web Search Results with Image Excerpts, *WWW 2008*, 21-30.

[22] Liesaputra, V., Ian, W.H., Seeking information in realistic books: a user study, *Proc JCDL 2008*, 29-38.

[23] Lin, S., Ho, J., Discovering Informative Content Blocks from Web Documents, *Proceedings of SIGKDD 2002*, 588-593.

[24] Liu, B., Grossman, R., & Zhai, Y., Mining Data Records in Web Pages, *Proc ACM SIGKDD Special Interest Group on Knowledge Discovery and Data Mining*, 601-606, 2003.

[25] Liu, Y., Zhang, D., Lu, G., Ma, W-Y., A survey of content-based image retrieval with high-level semantics, Pattern Recognition 40(1) 2007, pp. 262-282.

[26] Mayer, R.E., Moreno, R., Animation as an Aid to Multimedia Learning, *Educational Psychology Review*, 14:1, March 2002.

[27] National Science Foundation, Discoveries, http://www.nsf.gov/discoveries/, last visited 01/28/2009

[28] New York Public Library, Digital Collections, http://www.nypl.org/digital/collections_images.html, last visited 01/28/2009.

[29] Paivio, A., Mental Representations: a Dual Coding Approach, Oxford, England: Oxford University Press, 1986.

[30] Reis, D. C., Golgher P. B., Silva, A. S., & Launder, A. H. F., Automatic Web News Extraction Using Tree Edit Distance, *Proceedings of WWW 2004*, 502-511.

[31] Song, R., Liu, H., Wen, J-R., & Ma, W-Y., Learning Important Models for Web Page Blocks based on Layout and Content Analysis, *Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations*, 6(2), 14-23, 2004.

[32] W3C, Document Object Model (DOM) Level 2 Core Specification, http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/, 2000.

[33] Wang, J., & Lochovsky, F. H., Data Extraction and Label Assignment for Web Databases, *Proceedings of WWW 2003*, 187-196, 2003.

[34] Wildemuth, B., Marchionini, G., Yang, M., Geisler, G., Wilkens, T., Hughes, A., Gruss, R., How Fast Is Too Fast? Evaluating Fast Forward Surrogates for Digital Video, *Proceedings of ACM/IEEE Joint Conference on Digital Libraries* 2003, 221-230.

[35] Woodruff, A., Faulring, A., Rosenholtz, R., Morrison, J., Pirolli, P., Using Thumbnails to Search the Web, *Proceedings of SIGCHI 2001*, 198-205.

[36] Yi, L., Liu, B., Li, X., Eliminating Noisy Information in Web Pages for Data Mining, *Proceedings of SIGKDD 2003*, 296-305.

[37] Yu, S., Cai, D., Wen, J-R., Ma, W-Y., Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation, *Proceedings of WWW 2003*, 11-18.

[38] Zhao, H., Meng, W., Yu, C., Mining Templates from Search Result Records of Search Engines, *Proceedings of SIGKDD 2007*, 884-893.