2005

ACWW: Adding Automation to the Cognitive Walkthrough for the Web (CWW)

Richard Brown
University of North Florida

ACWW: ADDING AUTOMATION TO THE COGNITIVE WALKTHROUGH FOR THE WEB (CWW)

by

Richard Brown

A thesis submitted to the Department of Computer and Information Sciences in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES

November, 2005

The thesis "ACWW: Adding Automation to the Cognitive Walkthrough for the Web" submitted by Richard Brown in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee:

Signature deleted

11/04/05

Dr. Arturo Sanchez
Thesis Advisor and Committee Chairperson

Signature deleted

11/04/05

Dr. Roger Eggen

Signature deleted

Nov 04, 2005

Dr. Yap Siong Chua

Accepted for the Department of Computer and Information Sciences:

Signature deleted

11/4/05

Dr. Judith L. Solano
Chairperson of Department

Accepted for the College of Computing, Engineering and Construction

Signature deleted

12/8/05

Dr. Neal Coulter
Dean of the College

Accepted for the University:

Signature deleted

12/23/05

Thomas Serwatka
Dean of Graduate Studies

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Cognitive Walkthrough for the Web (CWW) is a Usability Evaluation Method that can be used as an approach to evaluate a website. Unfortunately, the original formulation of CWW has been applied by researchers in the area as a set of a semi-manual processes, which have proven to be extremely time-consuming, and as with any semi-manual process, error-prone. This thesis presents a web-based approach to automate CWW, which we call ACWW. The automation of these processes implies, on the one hand, that researchers can reduce the amount of time dedicated to computing the results associated with performing various analyses. On the other hand, it implies that the accuracy of computed results can be improved. ACWW is available from http://autocww.colorado.edu/~brownr/. Finally, ACWW has been mentioned in a paper by Dr. Marilyn Blackmon, a leader in usability research, at the Association for Computing Machinery (ACM) Conference on Human Factors in Computing Systems (CHI) in April of 2005 [Blackmon05].

Chapter 1

THE PROBLEM

An increasing number of people are using the World Wide Web for their standard information retrieval needs. Typical users can sit at home and have at their fingertips a volume of information which is several orders of magnitude larger than what is available to them in their local libraries. Thus the complexity of finding information becomes harder each year due to the exponential growth of the number of web pages available. When looking for information, users are faced with many potential options. It has already been shown that the inability of users to find the information they seek is one of the most frequent problems they face when using the World Wide Web [Pitkow97].

ACWW, the topic of this thesis, is our automated approach to an existing Usability Evaluation Method (Cognitive Walkthrough for the Web - CWW). Usability Evaluation Methods (UEMs) exist to identify and potentially correct any usability issues associated with interactive applications such as websites. Hertzum and Jacobsen

studied UEMS and discovered what has been called "the evaluator effect." That is, multiple evaluators evaluating the same interface and using the same UEM will arrive at different conclusions [Hertzum03]. The problem lies in the subjective evaluation of the interaction between the user, the task, and the interface [Hertzum03]. They also pointed out that the evaluator effect is independent of the experience of the evaluator. Adding evaluators helps to a certain point; but increasing the number of investigators promotes the diminishing of accuracy.

The Cognitive Walkthrough for the Web (CWW) has removed the subjectivity problem by utilizing Latent Semantic Analysis (LSA) (discussed in Chapter 2). CWW relies on LSA to objectively predict the user's action. A CWW analysis of an interface will yield the same results repeatedly.

Unfortunately, using CWW to analyze a single, small webpage with one goal requires about 28 minutes. This is due in part to the number of manual steps and the amount of copying and pasting required to perform the analysis.

This thesis presents an approach to automate CWW which leads to considerable improvements in performance and

consistency. For instance, our results show that ACWW provides a five fold increase in user performance over manually performing CWW.

In the manual process, the users have to copy and paste their data and results from various webpages. In addition, the users make the decision as to which heading/links are competitive or confusing. The consistency of the result depends on the users. Performing CWW manually has several points of possible failure, such as the amount of copying and pasting the user must perform. Also, while there is a standard file format for the results, the user must paste and organize the results in this manner. In both instances, it is easy for the user to miss an item or a result. Finally, if the user is not diligent in saving the parameters of the analysis with the results, the results can easily become meaningless.

By using ACWW, the users only have to submit their data and they will receive the results through their email. Compared to the process of manually performing CWW, ACWW only has one point of failure, the user entering the data. ACWW is an automated process and will produce the same results each and every time.

CWW is research in progress. As more discoveries in CWW are made, ACWW will have to be expanded. As such, ACWW will be broken into three modules so that new developments in CWW can easily be incorporated in ACWW or a replacement module can be developed. This will allow ACWW to grow as CWW develops.

The organization of the thesis is as follows. In chapter 2, we discuss the background information associated with our approach and outline competitors to CWW. Requirements gathered from the user are presented in Chapter 3. Chapter 4 discusses the Hardware and Software Architecture that is required to deploy the system we implemented. Chapter 5 delves into the design and implementation of our system. Chapter 6 documents its testing and the associated analysis of results. This thesis ends with conclusions and recommendations, a list of the literature we used, and appendices showing elements of the system we built.

Chapter 2

BACKGROUND ASSOCIATED WITH OUR APPROACH

2.1. Information Foraging Theory

The understanding of how a person accesses and understands information is necessary in order to achieve the goal of improving information access. In 1999, Peter Pirolli and Stuart Card put forth the theory of Information Foraging. The goal of the theory is to explain how information seekers gather and consume the information they encounter. Often, the information seekers navigate through various sources of information and must pick and choose which paths to follow and which to ignore. Information seekers try to pick up what is referred to as "information scent" to find their goal. Information scent is the (imperfect) perception of the value, cost, or access path of information sources from proximal cues, such as bibliographic citations, WWW links, or icons representing the sources [Pirolli99]. The proximal cues are not independent of each other. Rather, the individual cues combine in a multiplicative manner [Pirolli99]. If the

information seekers lose the scent, they tend to backtrack until they pick up the scent again.

2.2. CoLiDeS, CW, and CWW

2.2.1. CoLiDeS

Muneo Kitajima, Marilyn Blackmon, and Peter Polson developed a UEM model titled CoLiDeS (Comprehension-Based Linked model of Deliberate Search). CoLiDes is a cognitive model, which regards the selection of an object on the screen as an outcome of a multi-step process:

(1) Parse a web page into subregions,

(2) Select an action "Attend-to Subregion,"

(3) Elaborate the Elements in the Selected Subregion,

(4) Select a few Objects in the Selected Subregion, and finally

(5) Select an Object-Action Pair for the Next Action [Kitajima05].

The authors based CoLiDeS on earlier models developed by Kitajima and Polson, and the entire series of their cognitive models follow the Construction-Integration (C-I) cognitive architecture [Kintsch98]. C-I is a comprehensive

theory of cognitive processes centering on the core processes of reading comprehension and action planning. CoLiDeS was deliberately based on C-I because reading comprehension and action planning are the primary cognitive processes used during webpage navigation.

An item is considered to be a screen object if it is meaningful to the user and it can be acted upon by the user. Web links, navigation elements, and paragraphs (among others) can be considered as instances of screen objects. On any given website, there exists an average of 100 to 200 possible screen objects. To reduce the screen objects to a more manageable set, they are grouped together into 5 to 10 top-level schematic objects. The layout and overall design of the webpage should guide the user's eye toward the various elements of the page and allow the user to parse the webpage appropriately [Tullis98]. A webpage design that has a consistent or familiar interface will help the user parse the objects more quickly.

After the webpage is parsed into the top-level schematic objects, users' attention management mechanism guides them to a particular screen object. They focus on a top-level group, and then consider the actual screen objects in the

focused-on group. If none of the screen objects within the top-level schematic object provides enough of a cost/benefit ratio to act upon them, users may switch to another top-level schematic object and try again.

While focused within the top-level schematic object, users must comprehend and compare the results of selecting an object for action. They do this by comparing the screen object with the information stored in their long-term memory. Pirolli and Card used information scent to describe how the objects are related; CoLiDeS also uses information scent but uses five independent factors to compute the scent [Kitajima05, pp. 5]:

(1) The degree of semantic similarity between the user's goal and the heading/link text.

(2) Whether there is an adequate level of relevant background knowledge for successfully elaborating the associated knowledge in the semantic space.

(3) Whether a word used in the heading or link text is a low-frequency term in the user's background knowledge.

(4) The frequency with which the user has encountered the screen object/widget or specific heading/link.

(5) Whether there is a literal matching, partial or complete, between the user's target goal and a screen object.

Through these factors, users act upon screen objects that best represent their goal. Upon arrival at the new page, the process is repeated. If the design of the website is consistent, less time will be spent on choosing the top-level semantic objects for the subsequent pages.

As mentioned earlier, CoLiDeS uses LSA to compute the degree of semantic similarity between two texts. LSA is a computational model for natural languages and a theory of knowledge acquisition. The associated model was created to solve "Plato's problem;" i.e., the problem of how people have more knowledge than they have explicitly learned. In examining this problem, Landauer and Dumais discovered that "a very simple mechanism of induction, the choice of the correct dimensionality in which to represent similarity in learning about the similarity of the meanings of words, produces sufficient enhancement of knowledge to bridge the

gap between the information available in local contiguity and what people know after large amounts of experience" [Landauer97, pp. 211]. Through rigorous testing, they were able to show that LSA simulated the knowledge gained by school children from reading. They also discovered that 300 dimensions is a good approximation to represent the word knowledge.

Today, there are a variety of semantic spaces available. Most of these spaces are constructed from the texts of Touchstone Applied Science Associates, Inc. (TASA) corpus. This corpus includes the various sources that would have been read by a person at a given age-grade level and includes 3^{rd}-, 6^{th}-, 9^{th}-, and 12^{th}-grade and college-level TASA semantic spaces for speakers of American English. There is also a college-level French semantic space.

LSA represents each word as a vector in a 300-dimensional semantic space. Each multi-word text is a composite vector computed from the vectors of the composite words. The smaller the distance between two semantic objects in the semantic space, the greater the similarity. The distance between the two objects is actually measured by a cosine value of the angle between the two vectors created by LSA.

The results lie between +1 for identical objects, 0 for unrelated, and -1 for opposite objects. CoLiDeS uses LSA to compare a user's goal to the label of the screen object. The screen object with a value closest to 1 best represents the user's goal and would indicate a probable screen object the user will act upon.

CoLiDeS uses two additional LSA measures to compute information scent using the five factors that are discussed above [Kitajima05, p. 5]. To assess the sufficiency of background knowledge - the second of the five factors for computing information scent - CoLiDeS uses whether the length of the LSA term vector of the link label text is adequate. Term vector length is a measure of how much knowledge is associated with the term. Short term vectors indicate insufficient background knowledge to understand the term, and low word frequencies indicate words that people represented by the semantic space are unlikely to know. To assess low frequency words - the third of the five factors for computing information scent - CoLiDeS uses the frequency of the term in the selected semantic space.

2.2.2. CW

In 2002, Marilyn Blackmon, Peter Polson, Muneo Kitajima, and Clayton Lewis published a new paper on the Cognitive Walkthrough for the Web (CWW). CWW uses CoLiDeS as an underlying model, but it also transforms the Cognitive Walkthrough (CW) for applications [Blackmon02], a widely-used usability evaluation method since its inception. CW is an evaluation process that can be used during the design and development of a computer-based system. It predicts how a user will learn a computer-based system through exploration. CW takes into account the user's background in order to analyze the user group's characteristics.

A single evaluator or a group of evaluators performs the CW. They start by selecting a task that the user will perform. Along each step to complete the task, the evaluator(s) give a success or failure story. The story must be believable and must account for the user's background knowledge and experience. Finding problems early in the design and development process can save time and money by reducing training and support costs.

2.2.3. CWW

CWW transforms CW by using LSA instead of a group of evaluators. The subjective judgments of usability experts may not be consistent or might not catch all the errors. CWW, through the use of LSA, yields more consistent results and provides a rigorous analysis of the entire webpage. CWW embodies the same set of questions from the original Cognitive Walkthrough for the design team (quoted from [Blackmon03], pp. 498):

> "Q1) Will the user try to achieve the right effect?
>
> Q2) Will the correct action be made sufficiently evident to the user?
>
> Q3a) Will the user connect the correct sub-region of the page with the goal, using heading information and her understanding of the site's page layout conventions?
>
> Q3b) Will the user connect the goal with the correct widget in the attended-to-sub-region of the page using link labels and other kinds of descriptive information?
>
> Q4) Will the user interpret the system's response to the chosen action correctly?"

Q1, Q2, and Q4 are retained from CW. The evaluator uses the information from Q3a and Q3b to answer these questions. Questions Q3a and Q3b are approached by the use of LSA and they are the most important questions from the set in regards to user navigation.

A tutorial of CWW can be found at http://autocww.colorado.edu/~blackmon/. The process starts by creating a 100-200-word user goal statement. The goal statement should represent the various personas of the users. A persona is a representation of a user's personality. After several goal statements are written, a semantic space must be chosen for each goal/persona pair. Finally, the analyst must choose the correct webpage for each goal/persona pair. If the user is more likely to select a page other than the correct page through analysis, then this is indicative of a problem that needs to be corrected.

The analyst then needs to elaborate the link and heading texts. When a user encounters a heading or link, several semantically similar words are activated. The analysis needs to take these semantically similar words into account. To elaborate the headings and links, the

researcher clicks on "Elaborate Links" at http://autocww.colorado.edu. A whole set of raw text can be submitted at one time. The results need to be saved to be copied into the term-to-term, One-to-Many Comparison.

Next, the analyst feeds the goal statement, headings and link labels to LSA. A document-to-document, One-to-Many Comparison is performed to give an estimate of the semantic similarity between goal statement and the heading and link labels. The analyst ranks the table of cosines returned by LSA from highest to lowest. The highest-ranking items are those the user is most likely to select.

Next, the analyst runs a term-to-term, One-to-Many Comparison using the elaborated text obtained earlier. The comparison returns the vector lengths for the one-word and two-word heading/link labels. Any one-word heading with a vector length greater than .55 will be familiar to the user. For example, The Occult is unfamiliar to first-year College Students with a cosine of .08, while Magic, Supernatural, and Spirits is an appropriate replacement with a term vector length of .88 [Blackmon2003].

Finally, the analyst looks for goal-specific competing headings/links. This type of analysis is performed since two headings/links may not be similar, but they do compete for the user's attention when looking for the fulfillment of a specific goal. We use the results of the document-to-document, One-to-Many Comparison to identify any problems. If a heading/link ranks higher than the chosen correct link, then it is a goal-specific competing heading/link and the problem must be resolved.

A recent re-analysis of data compiled from past CWW experiments has resulted in a multiple regression model of problem difficulty [Blackmon05]. A multiple regression analysis was reported in Blackmon's original paper "Cognitive Walkthrough for the Web." This analysis helped develop the original model. In 2005, a new multiple regression analysis of new data providing an independent dataset of 64 additional items was performed. The model now explains 53% of the variance for 164 items. The formula, given below, predicts the number of clicks to complete an item and is derived from a third multiple regression of the combined dataset of 228 items [Blackmon 05].

Predicted Mean Clicks (PMC) = 2.199 + 1.656 (if correct link is unfamiliar) + 1.464 (if correct link has weak-scent) + (.0754 * Number of competing links nested under competing headings) + (0 * Number of competing links nested under correct headings) + (0 * Number of competing headings)

The algorithm to calculate the Predicted Mean Clicks has evolved through experimentation. In previous versions of the formula to calculate PMC, other coefficients were used in place of the 0's in the above formula. The formula's coefficients changed while creating ACWW and they may change again. Therefore, it was decided to allow the user to change the coefficients as needed.

2.3. Competitors of CWW: Other Website UEMs

2.3.1. WUFIS

Ed Chi, Peter Pirolli, Kim Chen, and James Pitkow developed two models that utilize information scent to predict users' navigational paths based on their informational needs. The models look at three areas: the entire website, the specific webpage, and the users. When examining the website, the models examine traffic flow and routes and how

hard it is to find information. The models answer the question of where the users come from and where they go when examining a webpage. The models also try to predict the interests of the user and the route the user will take based on these interests. Next the models compare the predicted path with the actual path a user takes.

The first model, WUFIS (Web User Flow by Information Scent) works by calculating the probability that a user will flow down a particular web link, given a specific information goal [Chi01]. WUFIS uses several matrices to represent the user's goal, the proximal cues, webpage topography, and link information. The proximal cues are the words in the link, the text surrounding a link, the graphics related to the link, and the position of the link on the page, among others [Chi01]. It uses a Term Frequency by Inverse Document Frequency as a key component of the algorithm to determine which webpages the user is likely to visit. Next, it uses a network flow algorithm to simulate users following the different links of a webpage. The result is a predicted usage log from which they can then infer the usability of a webpage.

The second model, IUNIS (Inferring User Need by Information Scent), examines a path already traversed by a user in order to determine the user's information goal. This model intuitively is the reverse of WUFIS, and it only constructs a keyword summary that is representative of the content in the user's path.

2.3.2. The Bloodhound Project

The Bloodhound Project is a refinement of WUFIS and is also built upon the idea of Information Foraging. The goal is to measure how easily a user can reach a selected webpage. The InfoScent™ Bloodhound Simulator uses automated methods to predict a user path through a website by examining the words surrounding the hyperlinks with a user goal [Chi03]. However, Morkes and Nielsen had previously found that users don't actually read the website, instead they scan it. Users tend to pick out individual words and sentences. In addition, highlighted keywords (hypertext) are more likely to be read by users [Morkes97]. Even so, Bloodhound was able to moderately correlate with two thirds of real user data.

2.3.3. SNIF-ACT

SNIF-ACT (Scent-based Navigation and Information Foraging in the Atomic Components Thought architecture), [Pirolli03, http://www2.parc.com/istl/projects/uir/projects/snif-act/], is based on Information Foraging Theory and ACT-R. Information on ACT-R can be found at http://act-r.psy.cmu.edu/.

The goal of SNIF-ACT is to model users such that it can simulate the user's web browsing patterns. ACT-R contains two major components: declarative knowledge and procedural knowledge. Declarative knowledge represents objects and events and how they relate. It is the "why." Declarative knowledge is represented as how likely it is needed at a particular time given a focus of attention.

Procedural knowledge represents the tasks necessary to reach an object or a goal. It is the "how." Procedural knowledge is represented as condition-action pairs. These pairs represent each step of cognition. When a conflict is created by one or matching pairs, the conflict needs to be resolved. Information scent resolves the conflict by distinguishing the best course of action.

The model uses the idea of spreading activation. The basic principle is that the user's goal and the website activate declarative knowledge. This activated knowledge spreads to other related knowledge which then becomes activated. When enough knowledge is activated to match against a known procedure, the user acts on procedural knowledge. By finding the highest amount of activity between the user goals and available links, a probability distribution can be created. This probability distribution is the likelihood of following different links on the website.

Psychologically, SNIF-ACT is an improvement over other systems, such as Bloodhound. SNIF-ACT is able to highly rank the same links the user chooses when fulfilling an information goal and generate good predictions. The model can also predict when the user will leave a website due to the diminishing of its information scent.

2.3.4. WebTango and WebCriteria

WebTango (http://webtango.ischool.washington.edu) deals mainly with graphic design issues. It does not take into account the user's information need. WebCriteria

SiteProfile (http://www.webcriteria.com) tries to predict a user's usage patterns on Design criteria. Like WebTango, it also does not take into account the user's information need.

2.3.5. Comparisons

The primary UEMs on which we have focused our attention are CoLiDeS-CWW, WUFIS, Bloodhound, and SNIF-ACT. Each of these uses the idea of information scent but they each use a different algorithm to detect the scent. Each claims to generate scent values that correspond well with user navigation.

The investigators of CoLiDeS-CWW have generated a large amount of empirical evidence to support their theory. The other UEMs have a very limited set of evidence. In addition, CoLiDeS-CWW uses a corpus of material that realistically represents the background knowledge of a particular user group. The others only try to compute the information scent and ignore the variability of each group's background knowledge.

CoLiDeS-CWW focuses on one page at a time while the others focus on a series of pages. However, each method has a high predictive rate.

The other methods look at each link on a webpage independently while CoLiDeS-CWW looks first at the different sub-regions of a webpage. This allows CoLiDeS-CWW to detect areas that compete for the user's attention. Also, CoLiDeS-CWW focuses on unfamiliar words that make up a heading or link. These can create navigation problems which the others do not detect.

The most important difference is that CoLiDeS-CWW is only partially automated while the others are almost fully automated. It is therefore important to increase the level of automation of CWW since it accomplishes results that the others do not.

We decided to investigate automating CWW for several reasons. First, the other models were already partially automated where CWW was not. Second, CWW has more empirical results over a multitude of experiments. Other models did not seem to have such rigorous testing of their models. Finally, the investigators associated with CWW

seem to have built a model for how the user interacts with websites rather than extracting the information from data collected. We believe that these elements make CWW a stronger model when compared to the other UEMs.

Chapter 3

CONTRIBUTIONS OF OUR APPROACH

3.1. Overview

The main objective of this thesis is to develop a method that reduces the amount of time it takes to perform a CWW analysis on a webpage, producing consistently accurate results. The secondary objective is to increase the ease of use in performing a CWW analysis. The benefactors are researchers who use CWW. They spend many hours performing a CWW analysis and hence greatly appreciate any improvement over the manual process. If the time to perform an analysis could be reduced, the researchers would better utilize their time developing and running new experiments. As more researchers have gotten interested in learning how to do a CWW analysis, it has become more important to make it quicker and easier for them to learn how to do CWW analyses. The primary researcher of CWW and the chief benefactor is Dr. Marilyn Blackmon. Dr. Blackmon is a Research Associate for the Institute of Cognitive Science at the University of Colorado. She has given considerable

feedback throughout all stages of ACWW and independently tested the output of ACWW compared to the more manual method for performing a CWW analysis at http://autocww.colorado.edu.

3.2. User Roles

The main user of ACWW is the website usability researcher. These researchers are mostly psychologists with little computer programming skills. The researchers will build a website or select an existing website for use in their experiments. The researchers may be investigating new techniques of CWW or may be analyzing the usability of an existing website.

3.3. Current Systems

To perform a CWW on a website, the user must be familiar with LSA and CWW. Papers on LSA can be found at http://lsa.colorado.edu. Papers on CWW as well as the tools to perform the analysis can be found at http://AutoCWW.colorado.edu and http://AutoCWW.colorado.edu/~blackmon.

The first step in starting a CWW analysis is to build a user goal statement. This statement should be 100-200 words in length, and it should represent the user's main goal and subgoal. The statement should be a narrative story the user would tell when describing their goal or a summary of what the user probably knows about a topic. After the goal statement has been built, the researcher would perform an "Unfamiliar headings/links" analysis on the webpage. Unfamiliar headings and links indicate that users are liable not to have adequate background knowledge of the topic. The researcher may also perform a "Low Frequency Words Analysis." Low frequency or unknown words cause the reader of the webpage a problem since they do not know the meaning of the words.

Next, the researcher would use the "Elaborate Links" tool. This tool expands the raw link/heading to include the words that reading the submitted link/heading would activate. The added words are both highly familiar and very similar to the raw link/heading text, simulating how skilled readers would elaborate the same text. This is a very time-consuming step, and it is suggested to limit the links to 50 in order to avoid browser timeout. Heading texts are paragraph-length texts built from the elaborated heading

text itself plus the elaborated texts of all their subordinate links.

The results from the "Elaborate Links" tools is then copied and pasted into the input of the "One-to-Many Analysis" tool for comparison with each user goal. Finally, the results of the One-to-Many Comparison are copied to an Excel spreadsheet.

Researchers currently use a specific format to save the results of the analysis. The text from the One-to-Many Comparison is copied into Column A and the resulting cosine is copied into Column B. The researcher sets Column C to indicate if the text is a heading or a link. The specific heading for each link is described in Column D. Columns F through G are a copy of Columns A through D sorted first by heading/link then by cosine value. The heading and link that leads to the correct webpage is highlighted in green. Competing links/headings are highlighted in yellow.

The last step is to calculate the predicted mean total clicks for each user goal. It is given by: Predicted Mean Clicks (PMC) = 2.199 + 1.656 (if correct link is unfamiliar) + 1.464 (if correct link has weak-scent) +

(.0754 * Number of competing links nested under competing headings) + (0 * Number of competing links nested under correct headings) + (0 * Number of competing headings). If a user must follow several webpages to reach their information goal, the predicted mean total clicks for each webpage are added together for a grand total.

The researcher may need to analyze multiple goals over multiple webpages, one goal over multiple webpages or multiple goals over one webpage. Following the current process, the work will become very tedious and time consuming. The researcher's workload is multiplied if the researcher needs to repeat the CWW analysis using two or more different semantic spaces, thereby simulating how people from different user groups would perform on the same task. In addition, the workload can be multiplied again if the researcher tries different parameters for some of the CWW analyses in order to find the parameters that offer the best fit to the data.

3.4. Production Considerations

Due to proprietary restrictions with the use of LSA, development occurred on a server at the University of Colorado. The final tool will reside on this server. Dr. Blackmon has given space at http://autocww.colorado.edu/~brownr/ for use with this thesis and has also installed the necessary PHP (http://www.php.net/) and MySQL (http://dev.mysql.com/) and Java (http://java.sun.com/) software for developing the ACWW tool on the server.

3.5. Functionality

ACWW takes as input the headings and links of the webpages, goal statements, correct heading/link designations for accomplishing the goal on the website, and analysis options. It then performs a heading/link elaboration and a one-to-many analysis on the data for each analysis option set provided by the user. The Predicted Mean Total Clicks will be calculated from the results of the analysis. All of the resulting data is saved into an Excel spreadsheet and emailed to the user. Our system is able to handle one

goal against one webpage, one goal against many webpages, many goals against one webpage, and finally many goals against many webpages. The final step also allows repeated analyses with the same dataset in different semantic spaces and with different parameters with virtually no additional effort by the researcher.

3.6. User Requirements

The main user requirement is that the analysis produces consistently accurate results and completes within 24 hours. The tool needs to conform to the same input standards as the current tools. This will allow the transition from the manual method to the new method easier. The new method needs to report the results in a similar manner to the old method.

3.7. User Centered Design

The design approach that will be used is the User Centered Design (UCD). UCD involves the user in all stages of the design process. The focus of the design is for the user to complete a set of tasks.

Regular input from the user drives the design. The first versions of the product are prototypes that have no real functionality. Users evaluate the prototype and give input about the design. In ACWW, a prototype was created for review. Dr. Blackmon ran through the prototype and gave her opinion and ideas on how to improve the interface. Subsequent prototypes incorporated the user's input and gained functionality. As the product progresses, it requires fewer changes to its design towards the end of product development since it accounts for user input from the beginning.

Rather than developing the internal architecture with team discussions focusing on code development, UCD focuses on the product from the user's perspective [Vredenburg02]. Meetings are geared to design and create a positive user experience.

Through development testing, if a user cannot complete a task that is a major component of the product, it is considered a problem. The product may work flawlessly, but if a user cannot figure out how to use it, they will move on to a product they can use. In the case of ACWW,

Dr. Blackmon had significant problems in one design. The section was redesigned several times in order to work out the problem until a satisfactory result was produced.

The user centered design process starts by deciding who is actually going to use the product and what need the product is going to fulfill. From the beginning, it was decided that ACWW would be used mainly by researchers. CW needs more research before it can be used by Web professionals. Next a prototype is drawn up and evaluated by a set of users. From there, changes are made and features added and another round of user testing is completed. The process of incorporating changes from the last test, adding features, and new testing is iterated until the product completed. ACWW followed this iteration until the end. It was tested through all stages, so when the first module was complete, we knew we did not have to revisit it, since it was thoroughly tested.

3.8. Cost Justifying Usability

It is important to keep in mind that benefits exist when developing with usability in mind. First, there is an

increase in user productivity. The product is developed with user efficiency in mind and allows the user to complete a task with a minimum amount of steps.

Another benefit is a decrease in training costs. The developed product will be easier to learn and, therefore, it will take less time to train a new user. A company pays the end user, the trainer, and for the training facilities. Reducing training time by even a few hours will result in substantial cost savings.

Without question, users make several errors a day. Some of the errors are made because of poor interface design. These errors may not become evident until a user sits down and uses the interface. Starting usability testing early in the design process will help find interface design issues that will cause problems with the users.

Imagine a product that goes into beta testing a problem surfaces for the first time. The problem is severe enough to cause most users to be unable to use the product. The design team now must go back and redesign the interface. Delays in shipping the product as well as retesting will become inevitable. Catching and fixing usability problems

early in the design will cost roughly one fourth of the cost of fixing the problem late in the design [Mantei88].

A hindrance to the use of CWW is the time it takes to perform the Walkthrough. A researcher who performs a Cognitive Walkthrough for the Web will spend several hours performing many repetitive steps. The purpose of this thesis is to create a tool that adds automation to CWW. It will reduce the amount of time it takes to perform the Walkthrough allowing researchers to spend more time researching and testing CWW.

Chapter 4

ARCHITECTURE

4.1. Hardware and Software Systems

The AutoCWW.colorado.edu computer is a Dell 530 workstation equipped with a dual 2.4-Ghz Xeon processor and 3.5GB RAM, running Red Hat Enterprise 4. The system runs Apache 1.3.23 as the webserver with PHP 4.3.2-25. The database system is MySQL 3.23.58-15. Finally, it uses PERL 5.84.

4.2. Flow of Control

Flow of control is an important concept associated with the architecture of a system. How information is gathered, processed, and given to the user dictates the design of the software.

To begin, the user enters information into the ACWW website. A website allows multiple users to use the system with no installation required on the user's system.

It allows the user to access it from multiple locations. Finally, changes to the system affect all users simultaneously without a need to distribute a new version.

After the user has submitted the information it is saved into the MYSQL database. The database was used due to the complexity of the data. Not only does the user enter the goal and website information, the user also defines the relationship between the two. The relationships are stored within the database (see Figure 1).

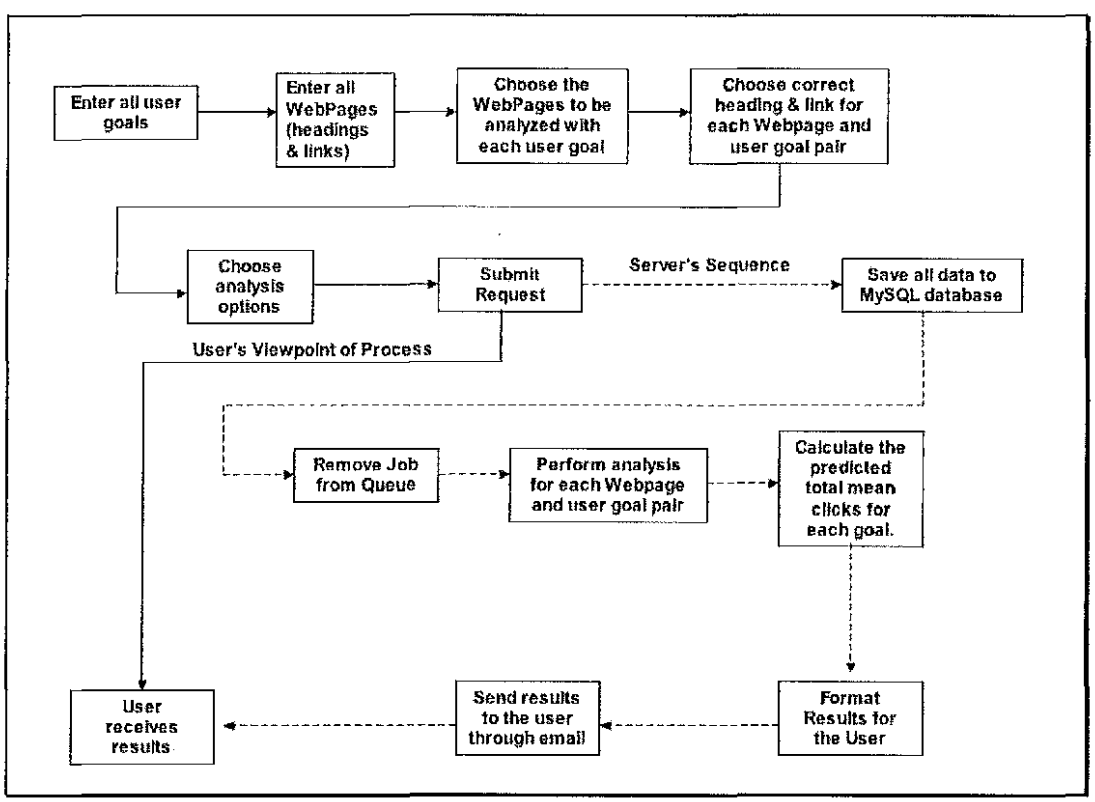


Figure 1: System's Flow of Control

After the information is loaded into the database, the back-end application retrieves the information as needed from the database and processes it. Results for each step are stored in the database in order to keep the relationships between webpages, headings, links, and goals intact. When the results are generated, Excel files are constructed from the results and emailed to the user. Finally, all computed information is persisted in the database.

4.3. Front-End

The front-end, or user interface, was implemented in PHP and HTML. The PHP file, ACWW.php, is one monolithic file that contains all the parts needed to run the ACWW interface. This file should reside in the root directory of the website.

There are several html files that also reside in the root directory. ACWW.php reads these files in a specific order and displays them on the screen as needed. This allows anyone to change the look and feel of the site without having to make changes to the PHP file. The order, size,

etc. of the forms can be changed as long as the name of the form remains the same.

ACWW.php does use session variables in order to store the information on the user side until all the information has been entered. At the last step, the data is organized and loaded into the database all at once. When the information has been loaded into the database, the session cookie is destroyed along with any information residing on the user's side.

ACWW.php uses a MySQL database for its database needs. If a different user name or password is needed to access the information, there is only one line in the script that needs to be changed. Finally, ACWW.php calls a java back-end application to perform the analysis. If the PHP file and its support files are moved, the directory information in the PHP file will also need to be changed.

ACWW was created with security in mind. The miscellaneous sub-applications written for ACWW are in the home directory rather than the public web directory. This prevents anyone from arbitrarily trying to run these applications. The database is used as an intermediary for the web application

and the back-end Java application. When the analysis is complete, the data and results are removed from the database.

If anything happens to the database, the database structure has been saved as a MySQL Query. This Query can be used to recreate the database if necessary. The MySQL Query is saved as a file and is stored in the home directory.

4.4. Database

As mentioned before, the system uses a MySQL database to store the information gathered from the interface, temporary analysis results, and the final results before the reports are generated. There is nothing special concerning the implementation of the database other than the constraint that speed was a much more important concern than space.

4.5. Back-end Application

For the back-end java application to run properly, additional files need to be installed on the server. The MYSQL java connector needs to be installed in the same directory as the class file. This allows the java application to connect to the MYSQL database. The current version can be found at http://dev.mysql.com/downloads/connector/j/3.0.html.

The java application calls several PERL and shell scripts in the parent directory in order to perform the analysis and email the results. The PERL Scripts are modified versions of the PERL SCRIPTS used in the manual process. The scripts were modified so that the data is returned to the back-end application instead of the user.

The supporting scripts reside in the parent directory and the class file resides in the website's root directory due to security concerns. With the support files one level higher than the website's root directory, web users cannot access or execute these files. Instead, only the application that is called by the interface can execute these support files.

The first file, acww-elaborate.cgi, performs the elaboration on any link or heading it is given. The next file, acww.one2many.cgi, performs the one-2-many analysis. These files are modified files from the existing AutoCWW website (http://autocww.colorado.edu). These AutoCWW files were modified in such a way to accept input from the command line as well as report only output back to the screen. The back-end application captures the output. The email-rep.sh script emails the reports to the user after compressing them. The base64 file is called by the email-rep.sh script to convert the compressed file into a format that can be included within the email sent to the user.

Chapter 5

IMPLEMENTATION

5.1. Design

ACWW is separated into three modules. First is the user interface, second is the database, and the third is the back-end application. The user interface can be replaced or a separate interface developed with no changes to the database or the back-end application needed.

5.2. User Interface

The user interface was designed in order to be as easy to use as possible while gathering all of the needed information. Compared to the existing process, more information is gathered from the user. This extra information is needed in order to perform more work than the old system.

To display all the options in ACWW with a simple example, this chapter will describe how a researcher would use ACWW to perform an analysis on two different webpages using two different goals. The analysis will predict how easy or difficult it will be for a user to select a correct link to accomplish each goal on one or more different webpages. The researcher would start by navigating to http://autocww.colorado.edu/~brownr/ACWW.php. ACWW prompts the researcher to complete a five-step process, supplying the information that ACWW needs to perform the analysis. At the end, researchers provide their email addresses to ACWW and submit their requests to perform the analyses. After ACWW performs the requested analysis, ACWW sends an email from Results@ACWW with an attachment named Results.zip. The attachment contains one or more Excel files that show the completed analysis.

Figure 2 shows the webpage for the first step in ACWW, the step for inputting one or more goal statements. The researcher can copy and paste – or type – the goal statement into the bottom text box.

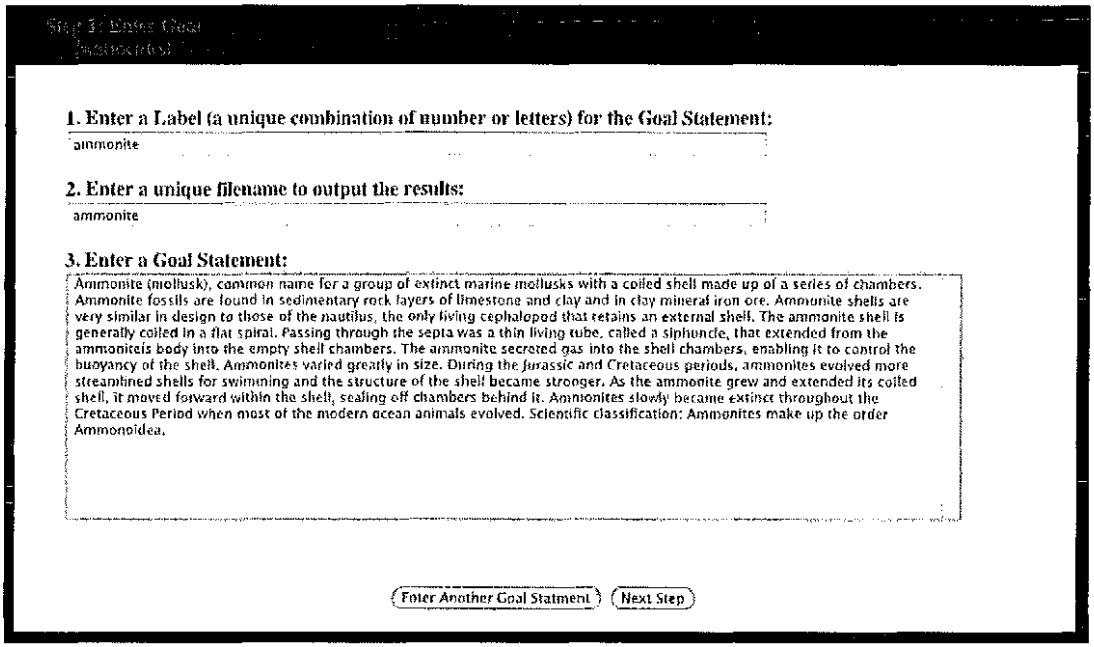


Figure 2: Entering a goal in Step 1 of ACWW.

In the top text form, the researcher needs to enter a label that will enable the researcher to easily identify the goal in a later step, where ACWW will ask the researcher to designate the correct heading(s) and correct link(s) for this goal.

In the middle box the researcher must enter a filename for the Excel file that ACWW needs in order to save the results from the analysis of this goal. The researcher would input the first goal and then select the 'Enter Another Goal Statement' in order to input another. When the researcher

has finished inputting all of the goal statements, he/she can select 'Next Step' in order to move on to Step 2.

Step 2 (c.f. Figure 3) consists of defining the headings and links within a given webpage and the relationships between the two. The researcher starts by submitting all of the headings within the webpage, as shown in Figure 3.

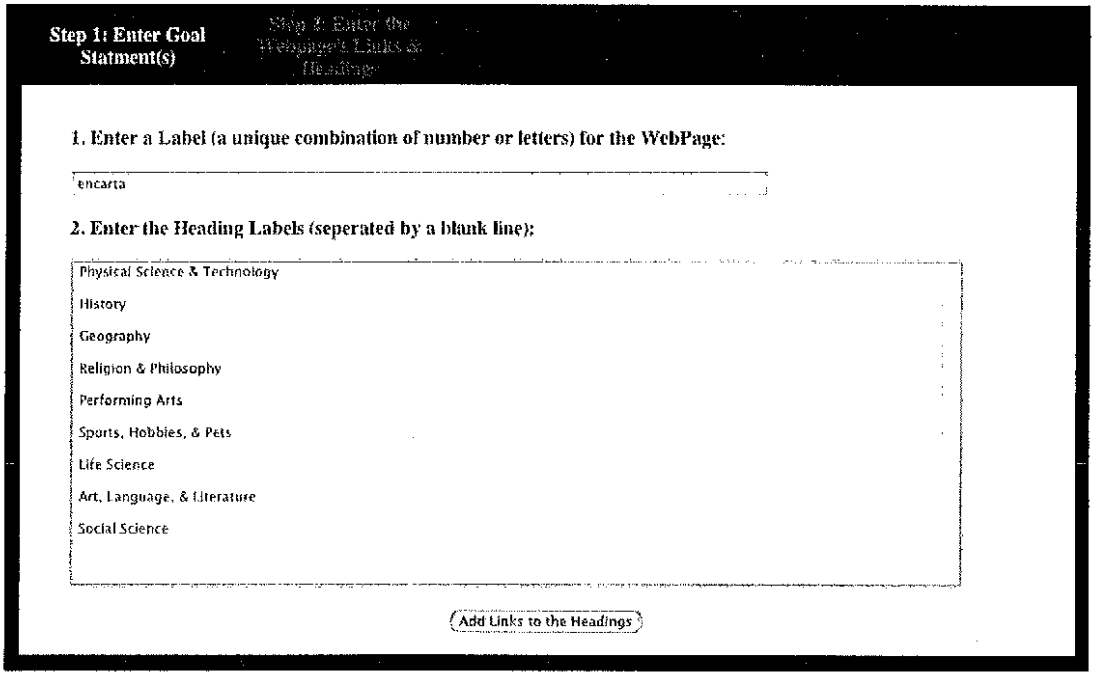


Figure 3: Entering the headings in Step 2 of ACWW.

After selecting, 'Add Links to the Headings' the interface loops through each of the headings, allowing the researcher to input the links for all headings. The user cannot

inadvertently skip any heading since the researcher has only one button to press, 'Enter Link Labels for the next Heading.' Figure 4 shows how a researcher would enter the link labels for one of the headings, 'Physical Science & Technology.'

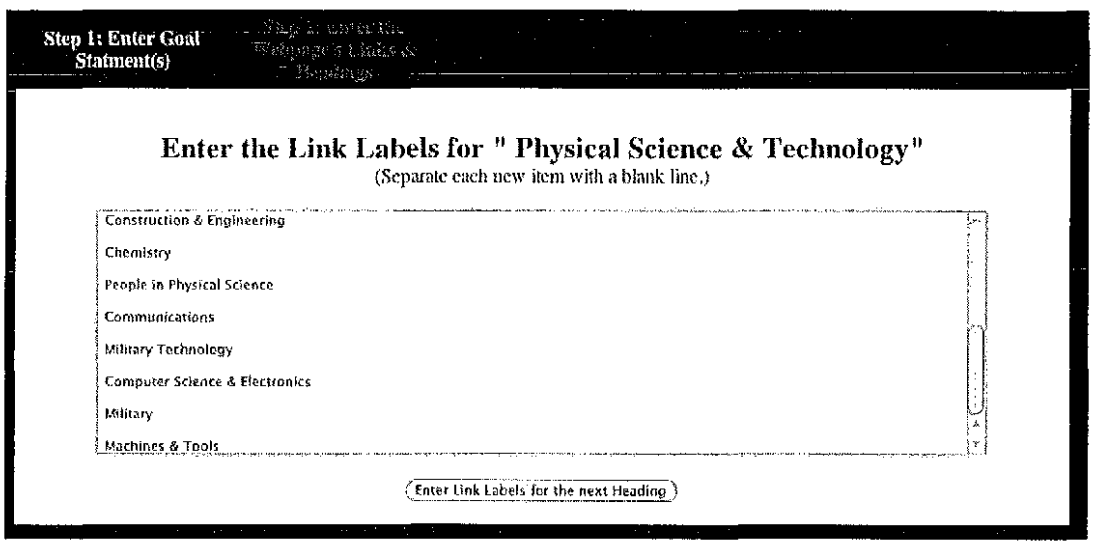


Figure 4: Entering the links for a heading in ACWW, Step 2.

On the last heading, ACWW presents the user with two new buttons, 'Enter the Headings and Links of Another Webpage' and 'Next Step'. These buttons are shown in Figure 5. If the researcher wants to enter another webpage, the researcher selects the first button, and Step 2 of ACWW starts over so that the researcher can add another webpage to be analyzed. If the researcher is done entering

webpages the researcher selects 'Next Step,' and then the researcher is taken on to Step 3 of ACWW.

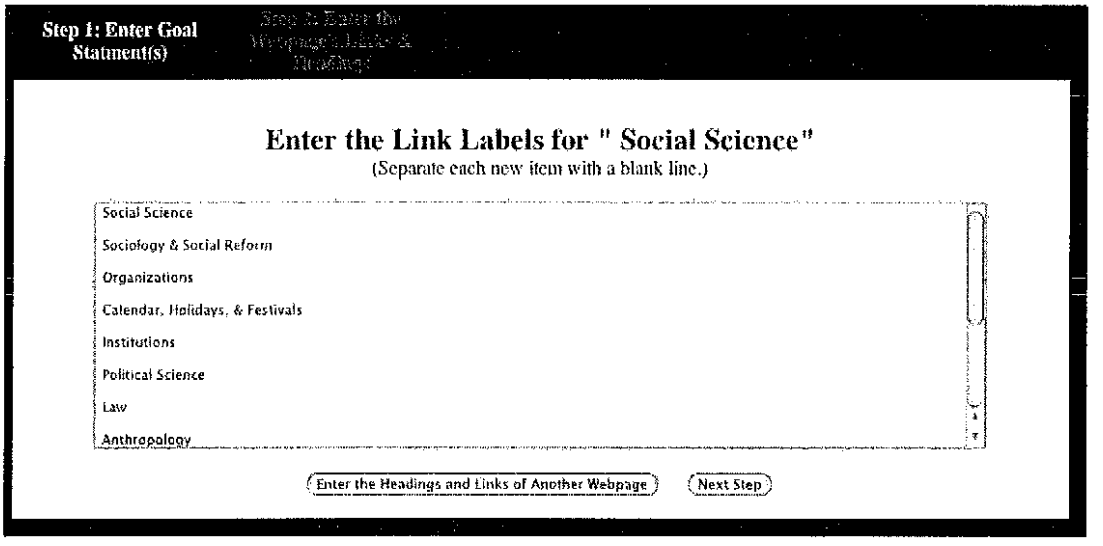


Figure 5: New buttons that appear with the form for entering the links for the last heading in ACWW, Step 2.

In step 3, the researcher needs to specify which goals to analyze with which webpage. The default, shown in Figure 16, sets ACWW to analyze all goals entered in Step 1 against all webpages entered in Step 2, but the researcher has the option to uncheck the boxes in order to perform just some of these analyses, not all.

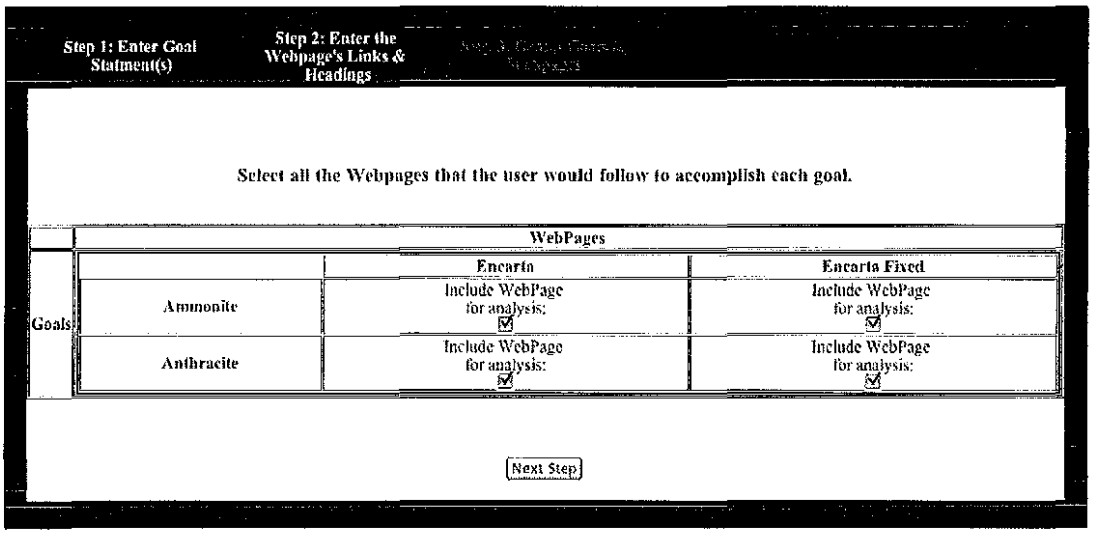


Figure 6: Step 3 of ACWW designates which goals and which webpages to include in the analysis.

If, for example, the researcher wanted to analyze the ammonite goal with the Encarta webpage and the Encarta fixed website, the researcher would leave the checks in the checkboxes. If the researcher only wanted to analyze the anthracite goal against the Encarta website, the researcher would uncheck the appropriate check box - see Figure 7.

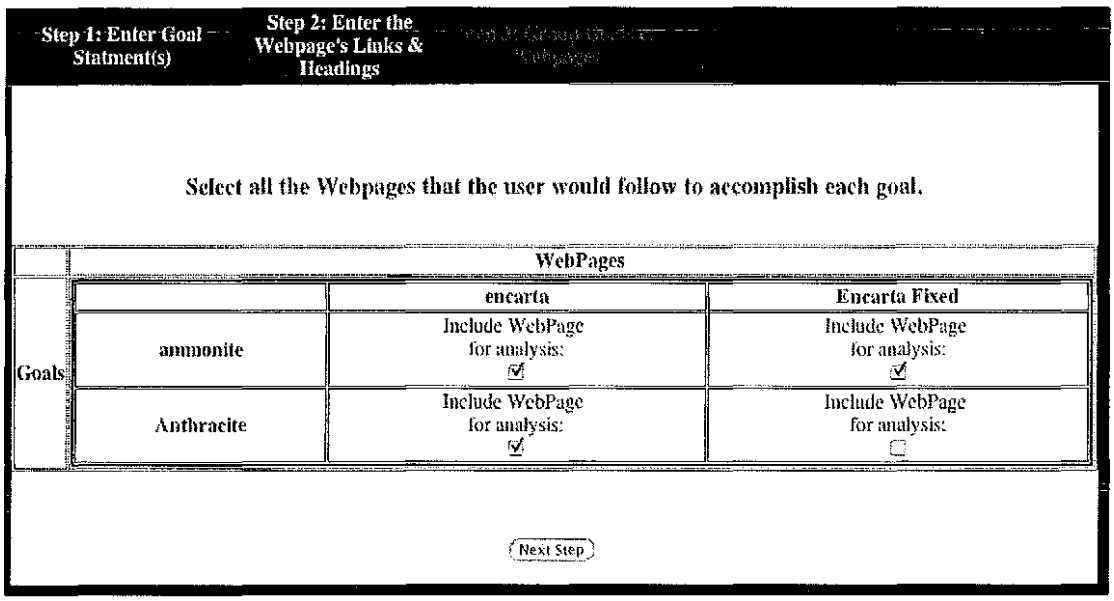


Figure 7: Overriding the default to do only the analyses desired, not compare all goals with all webpages.

After selecting 'Next Step' to complete Step 3 the researcher moves to step 4. In Step 4, ACWW will cycle through all of the goal/webpage combinations selected by the researcher. The user needs to define the correct heading and link for the goal. The links associated with a goal are offset underneath its heading. The researcher needs to select the appropriate heading/link and click on 'Next Step,' as illustrated in Figure 8.

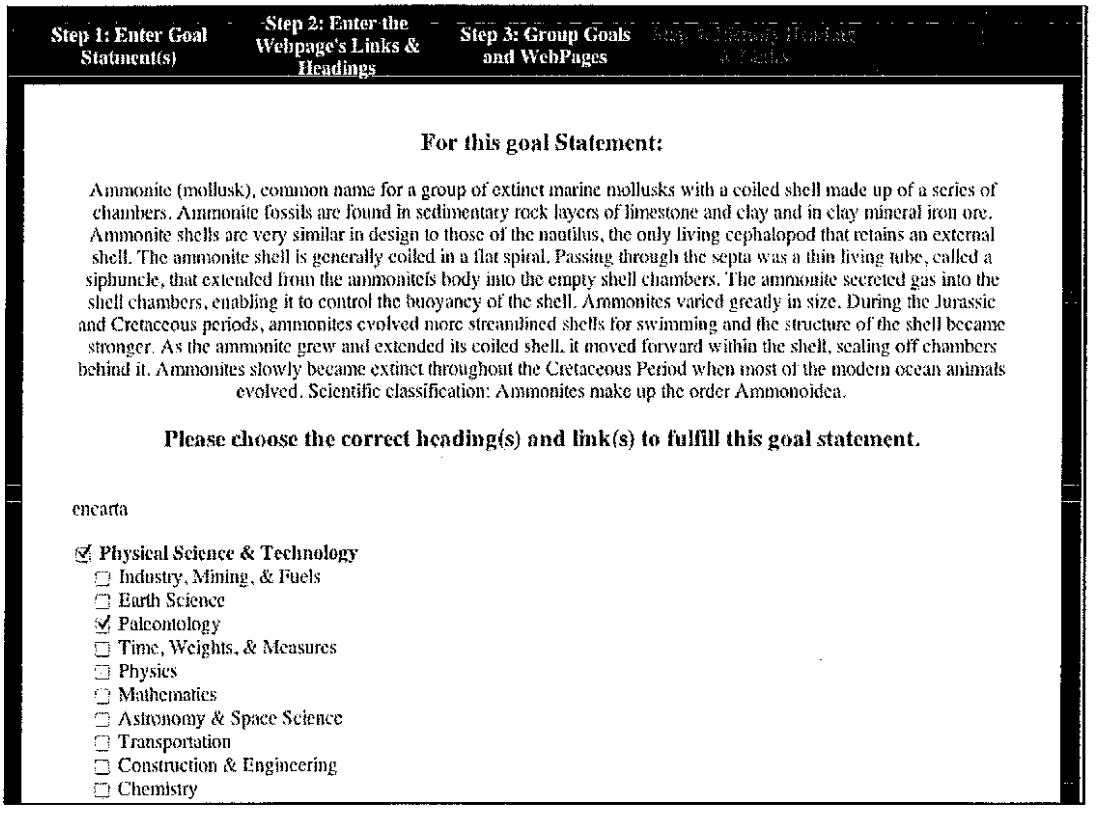


Figure 8: ACWW Step 4: designating the correct heading and link for one goal.

In the final step, Step 5 shown in Figure 9, the researcher selects the semantic space and the options for elaborating the heading and link texts. The user can also change the default values of the predicted mean clicks formula, if desired for research purposes.

A powerful final option of ACWW for Step 5 is found in the 'Enter another set of options to use with the Goals/Webpages entered' button (see Figure 9). This will save the options and present the same page again. When

ACWW runs, it will perform a full analysis on every goal/webpage combination for every set of options the user submits. The researcher can easily have ACWW run from two to 20 or more different analyses on the same goal/webpage with only minor differences in the options.

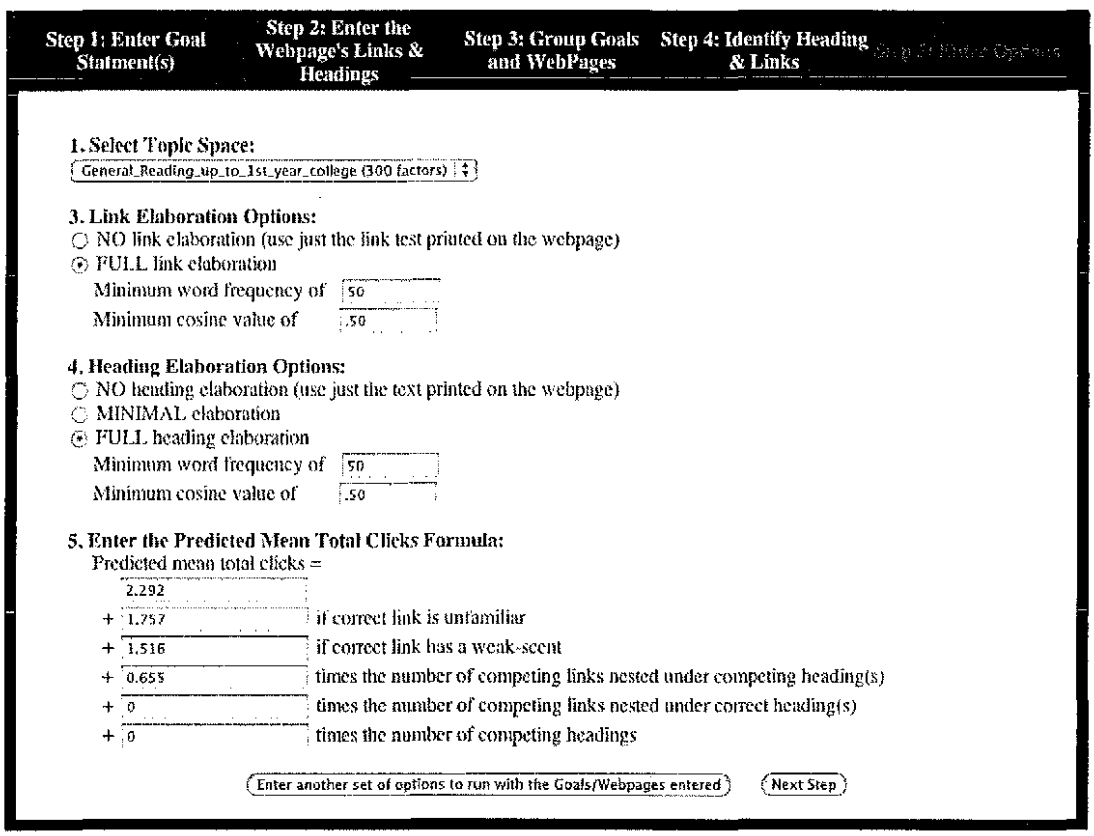


Figure 9: ACWW Step 5: Selecting the options for one or more repetitions of the analysis.

When the researcher selects the 'Next Step' button shown in Figure 9, the researcher is asked to enter their email address. This email address is where the Excel files will

be emailed. Depending on the size of the analysis to be performed, the user can expect results in a couple of minutes to several hours.

5.3. Database

ACWW uses MySQL to store the information gathered from the interface. It was chosen because several programming languages can utilize the power of MySQL. This allows flexibility in designing the interface and the back-end application.

Early on in the design process, it was decided that once the results had been emailed to the user, the data would be deleted from the database to avoid filling up limited space on the server. At that point, it was recognized that the database should be built for speed with only minor concern about the amount of space it utilizes.

Both the user interface and the back-end application start at the JOB_OPTIONS table when they need to retrieve or store data. This table, shown in Figure 10, stores

everything in the last two steps of the user interface as well as the unique identifier for that particular analysis.

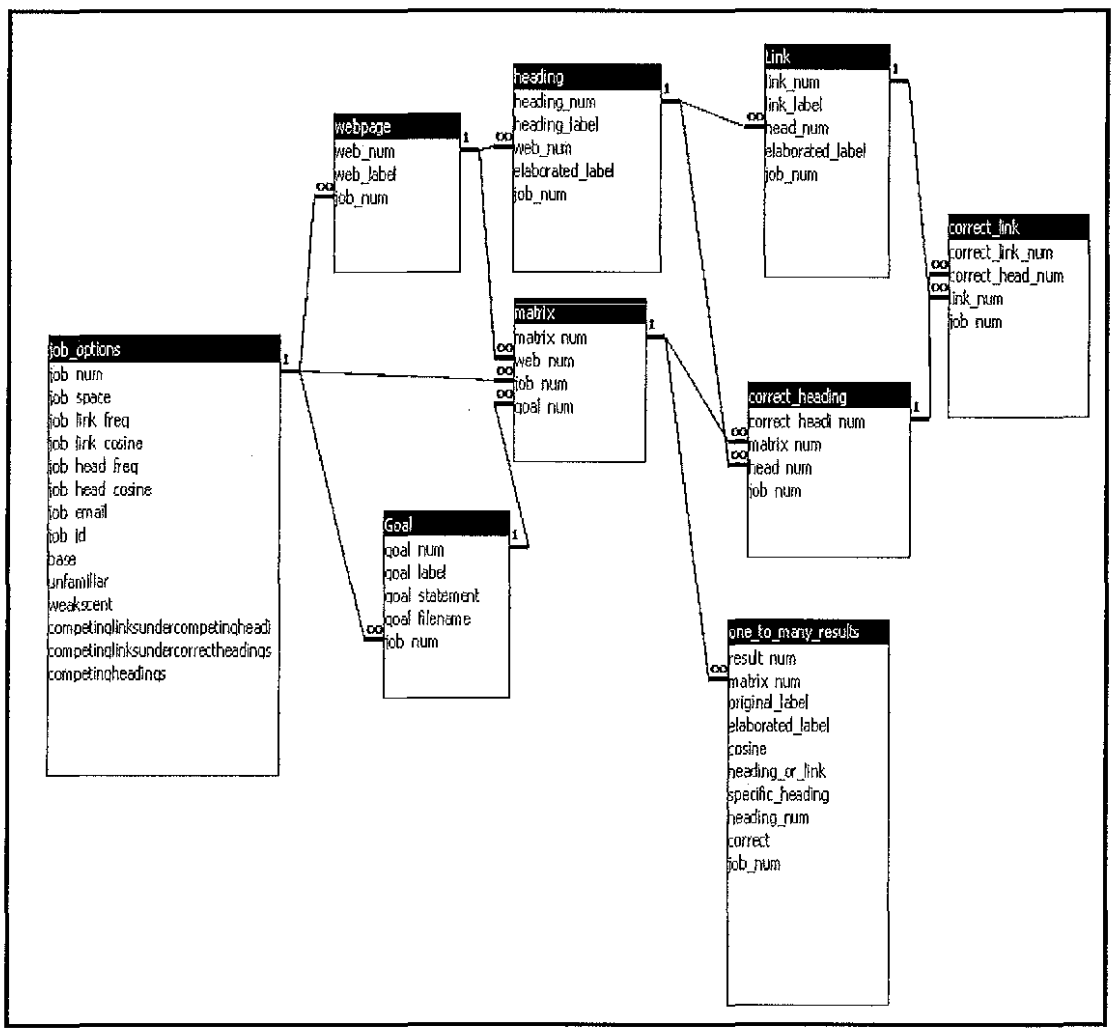


Figure 10: Our data model

The next table, GOAL, contains all of the information the user entered for each goal. There is a one-to-many

relationship between JOB_OPTIONS and GOAL since one analysis can have many goals.

The WEBPAGE table consists of the webpage label and its number. Since many webpages are associated with one analysis there is a one-to-many relationship between JOB_OPTIONS and WEBPAGE. One webpage can contain many headings giving another one-to-many relationship between WEBPAGE and HEADING tables. The heading table contains the raw heading label and will contain the elaborated label when the elaboration is complete.

When a user submits different options on the second to last step of the interface, the elaborations may change due to different heading frequencies and cosines chosen. Hence, the reason ACWW has duplicate copies of the goals and webpages for each item in the JOB_OPTION table.

Each heading contains one or more links underneath. The LINK table has a one-to-many relationship with the HEADING table. The LINK table contains the raw link label and will contain the elaborated label when the elaboration is complete.

When the user chose the goals to analyze with the different webpages, a simple matrix was created to keep track of them. When the data was stored into the database, the matrix was stored in a table called MATRIX. The table has a one-to-many relationship with WEBPAGE, GOAL, and JOB_OPTIONS. The table only has foreign keys that point to the specific webpage and goal that should be analyzed next for the current analysis.

For each item in the matrix, there is an associated correct heading and link. The relationship between the heading and link needs to be preserved so we know which heading a link is under. There can be more than one correct heading. There can be more than one correct link under the same heading or a different heading. Two tables keep track of all of this data. CORRECT_HEADING has a one-to-many relationship with MATRIX and CORRECT_LINK has a one-to-many relationship with CORRECT_HEADING. These two tables keep track of which matrix they belong to as well as point to the correct heading/link respectively.

The last table, ONE_TO_MANY_RESULTS, stores the data from a one-to-many analysis. Since space is of little concern, data from LINKS and HEADINGS is duplicated within this

table. This allows for faster processing within the back-end application. When the reports are generated, almost all of the data is pulled from this table.

5.4. Back-end Application

For a lack of a better name, the back-end application was simply named backend.java. It is the piece of software that runs after the user has entered all of the information and it is saved in the database. The user can go do whatever they wish and come back later to retrieve their results via email. The application runs in the background quietly running the analysis the user requested.

Backend.java was written in Java since it is a portable language and has MySQL support. If someone else designs a new interface and wants it to run on a different system, they only need to provide the MySQL database and Java support.

The application was written modularly and can be described in 4 modules. The first is the heading/link elaboration.

The next module is the one-to-many analysis followed by reporting. Database cleanup is the last module.

The first module loops through each webpage specified in JOB_OPTIONS and iterates through each heading. It performs an elaboration on heading then iterates through each link under the heading. A PERL script written by Dipti R. Mandalia for the original CWW user interface (see http://autocww.colorado.edu) performs the elaboration. This script, part of the old process, was modified to work with ACWW. ACWW gives the PERL script the heading/link, cosine, and frequency through a command line. The script then returns the elaborated text. The elaborated text is stored with the raw text for future processing needs. If no elaboration is required, the raw text is saved into the elaboration text field.

The second module performs similarly to the first. It iterates through each heading and link in a webpage and performs a one-to-many analysis for each item. A PERL script written by Dipti Mandalia for the original interface (http://autocww.colorado.edu) completes the one-to-many analysis. ACWW simply feeds the needed information and the

script returns the cosine for the heading/link. All the information is stored within the ONE_TO_MANY_RESULTS table.

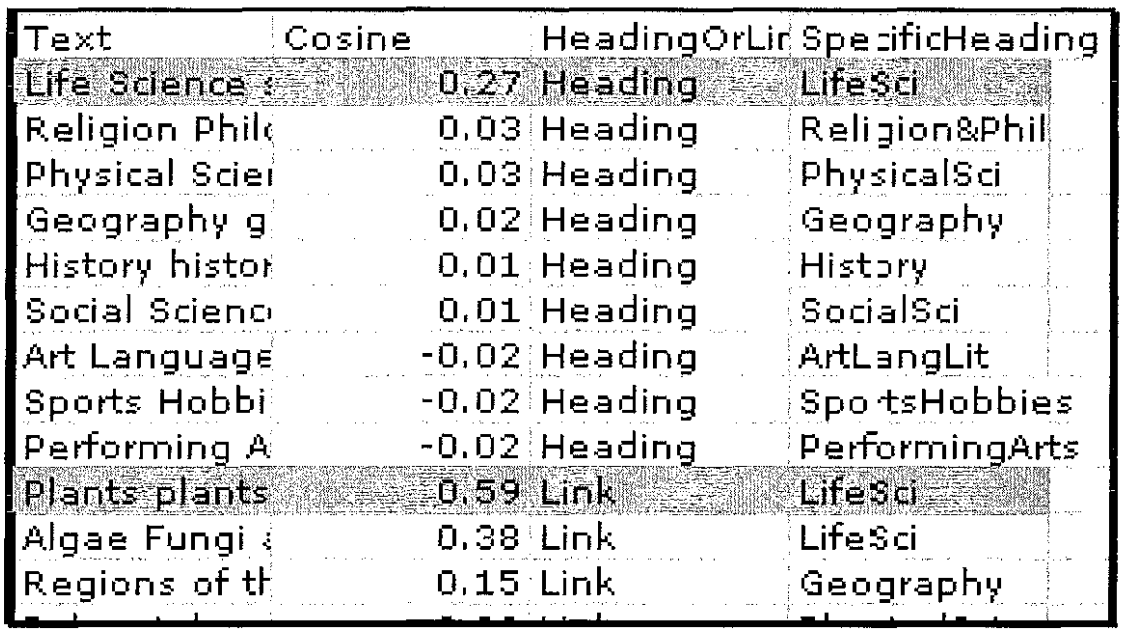
The third module has two steps. The first step is to calculate the Predicted Mean Clicks (PMC) for the goal. The data is pulled from the ONE_TO_MANY_RESULTS table and compared against the rules listed in Appendix A. Any heading or link that matches any of the rules is flagged for each rule it matches. Some items may match several rules at the same time. The PMC is then calculated using the coefficients defined by the user in the interface.

After the Predicted Mean Clicks is calculated, the system sorts the data first by whether the item is a heading or a link, then by the cosine value. An Excel file is generated for each goal. These Excel files are compressed into a zip file that is emailed to the user as an attachment. If the user submitted multiple job options, the user would receive multiple emails, one for each job option submitted.

Finally, the system goes through and removes all data from the database for the analysis that was just performed. This allows us not to concern ourselves with space, but rather performance.

5.5. Preliminary Testing

One of the goals of ACWW was to increase the efficiency of the system, thereby freeing up more time for the researcher. To that end, ACWW not only recreates the same information but it also produces even more information compared to the manual process. Figure 11 is a screen shot of the Excel file generated by the old method. Text is the elaboration of the original text. Cosine is produced by the one-two-many analysis. Complete details on how the file is produced can be found in the AutoCWW tutorials at http://autocww.colorado.edu/~blackmon/.



Text	Cosine	HeadingOrLir	SpecificHeading
Life Science :	0.27	Heading	LifeSci
Religion Phil	0.03	Heading	Religion&Phil
Physical Scie	0.03	Heading	PhysicalSci
Geography g	0.02	Heading	Geography
History histor	0.01	Heading	History
Social Scienc	0.01	Heading	SocialSci
Art Language	-0.02	Heading	ArtLangLit
Sports Hobbi	-0.02	Heading	SportsHobbies
Performing A	-0.02	Heading	PerformingArts
Plants plants	0.59	Link	LifeSci
Algae Fungi :	0.38	Link	LifeSci
Regions of th	0.15	Link	Geography

Figure 11: Report created by a researcher

is a screenshot of a report generated by ACWW. The layout and meaning of the highlighted text is similar to the previous layout. However, additional information is provided by ACWW that would have taken considerable effort by the researcher to produce. The first row is the user provided goal followed by the calculated Predicted Mean Clicks for the specific website. The report not only shows the result, but also the full equation that produced the results.

Goal: Ammonite (mollusk), common name for a group of extinct marine mollusks with a coiled shell made up of a series of chambers. Ammonite fossils are

Predicted Mean Clicks = 2.292 + 1.757 (Link is unfamiliar: true) + 1.516 (Link has a weak-scent: false) + 0.655 * 3 (Number of competing links nested unde
6.014

Heading Frequency: 30.0 Heading Cosine: 0.5
Link Frequency: 30.0 Link Cosine: 0.5
Space: tasaALL
Webpage: Encarta

Original Label	Text	Cosine	Term Vector	Heading Or Link	Specific Heading	Correct	Weak-Scent Correct Link	Unfamiliar Correct Link	Competing Link under Competing Heading	Competing Link under Correct Heading	Com Heac
Life Science	life science	0.18	2.59	Heading	life science s						X
Physical Scienc	physical sc	0.1	3.79	Heading	physical scien	X					
Art, Language,	art languag	0.07	4.06	Heading	art language l						
Social Science	social scien	0.04	3.15	Heading	social science						
History	history hist	0.04	1.2	Heading	history histor						
Performing Art	performing	0.03	0.73	Heading	performing art						
Geography	geography	0.02	0.5	Heading	geography geog						
Sports, Hobbie	sports hob	0.01	1.32	Heading	sports hobbies						
Religion & Phil	religion phi	0.01	1.79	Heading	religion philo						
Periods & Style	periods sty	0.27	0.91	Link	Art, Language,						
Invertebrate A	invertebrat	0.23	2.96	Link	Life Science				X		
Paleontology	paleontolo	0.22	0.06	Link	Physical Scien	X		X			
Anatomy & Phy	anatomy p	0.21	0.51	Link	Life Science				X		
Industry, Minin	industry m	0.2	2.69	Link	Physical Scien					X	
Mammals	mammals	0.2	1.65	Link	Life Science				X		
Oceans & Seas	oceans sea	0.19	1.87	Link	Geography						
People in Life	people in li	0.11	3.29	Link	Life Science						
Reptiles & Am	reptiles am	0.11	2.29	Link	Life Science						

Figure 12: ACWW generated report

Next, the analysis settings are reported. During testing, we encountered a problem of misinformation. Results of a previous test by performing CWW manually were used to compare to the results ACWW generated. Unfortunately the results were not matching. ACWW seemed to be performing fine and no problems could be discovered. We then scrutinized the results from the manual method. It turned out that the analysis settings were different than originally thought. The reason was that the analysis settings from the manual method were stored separately from the results. Once the results were separated from the directory, there was no easy way to determine the values used. It is easy to mistakenly give the wrong information when sending a file. After this problem arose, it was decided that ACWW would add the analysis settings used into the results file. This would ensure that the results would never be separated from the analysis settings used.

After the analysis options are reported, the results are listed. The first column (see Figure 12) is the original text given by the user. The next column is the resulting elaboration of the original text using the values given by the user. Next is the resulting cosine and term vector from the analysis. After the term vector, the original

text is classified as a heading or link, and the next column indicates its specific heading. Next, the correct heading and link is marked.

The next set of columns flows from the automated rules in Appendix A. ACWW will mark any item as a competing heading, competing heading competing link, correct heading competing link, weak scent correct link, and an unfamiliar correct link. A link may be marked as having any one of these specific problems. This information does not exist within the manual method. It was added to the results since it gives valuable information to the researcher. A researcher can use this information to fix the heading/link.

Chapter 6

TESTING AND EVALUATION OF RESULTS

6.1. Testing

The system was tested for accuracy as development occurred. This allowed any issues to be fixed as the problem arose. At the end of development we ran the system through multiple analyses and compared them to the results achieved by Dr. Blackmon. The results matched perfectly. Dr. Blackmon then tested the system for performance issues. She compared the length of time to analyze 3 types of webpages: small, medium, and large.

We defined the size of a webpage according to the number of objects on the page. The average number of hyperlinks on a Webpage is 56.1 [Koehler99] with a standard deviation of 21.8. We defined a small website as -2 standard deviate from the mean (35 links or less), a medium website as 1 standard deviation from the mean (36 to 76 links), and a large website as +2 positive standard deviations from the mean (77 links or more).

6.2. Results

The results from Dr. Blackmon's tests are shown below in Table 1, with the results listed in minutes. In each case, there was a significant decrease in the amount of time to perform an analysis. As the complexity of the analysis increased, there was a more significant savings between the two methods.

The first case with one goal and one small webpage resulted in a time difference of 23 minutes. The largest test, consisted of 64 goals with 3 webpages, resulted in a time difference of 1023 minutes. For small and large projects, the researcher would gain significant time savings by using the new method over the old.

Small webpage		One goal, one small webpage	One goal, 3 small webpages
	Manual	28.482	85.446
	ACWW	5.1	8.592
	Difference	23.382	76.854
	% Difference	558.47%	994.48%
		64 goals, one small webpage	64 goals, 3 small webpages
	Manual	251.061	753.183
	ACWW	59.756	113.033
	Difference	191.305	640.15
	% Difference	420.14%	666.34%

Table 1: Results

Medium webpage		One goal, one medium webpage	One goal, 3 medium webpages
	Manual	41.342	124.026
	ACWW	6.333	17.6
	Difference	35.009	106.426
	% Difference	652.80%	704.69%
		64 goals, one medium webpage	64 goals, 3 medium webpages
	Manual	264.921	794.763
	ACWW	79.1	185.49
	Difference	185.821	609.273
	% Difference	334.92%	428.47%

Table 2: Results (Continuation)

Large webpage		One goal, one large webpage	One goal, 3 large webpages
	Manual	82.334	247.002
	ACWW	7.45	21.03
	Difference	74.884	225.972
	% Difference	1105.15%	1174.52%
		64 goals, one large webpage	64 goals, 3 large webpages
	Manual	400.484	1201.452
	ACWW	81.146	177.75
	Difference	319.338	1023.702
	% Difference	493.54%	675.92%

Table 3: Results (Continuation)

In Table 4: Savings, the total savings between all of the tests can be seen. On average, ACWW is 5.61 times faster than the old method. This is a huge improvement and is a big step forward for more research into CWW.

Total Manual Time	4274.496
Total ACWW Time	762.38
Savings	3512.116
Times Faster	5.61

Table 4: Savings

Figure 13 below shows the relationship between the size of the website and number of goals compared to the performance difference between ACWW and the current process. If there was no performance increase, there should be a flat line at 0.

The first three sets are very close to each other. The last set, 64 goals with 3 webpages, gave a sharp increase in the amount of time saved. The reason is that adding an extra goal to an analysis is very easy and not very time consuming. However, adding extra webpages is very time consuming and results in the bulk of the time a researcher will spend using the current process. As the complexity and the number of webpages grow, so will the amount of time the researcher spends entering the data for analysis. This is most evident when looking at the performance between a medium size website and a large website.

ACWW increases the performance by minimizing the amount of data the user needs to enter. The same goal(s) is used across multiple webpages resulting in substantial time savings.

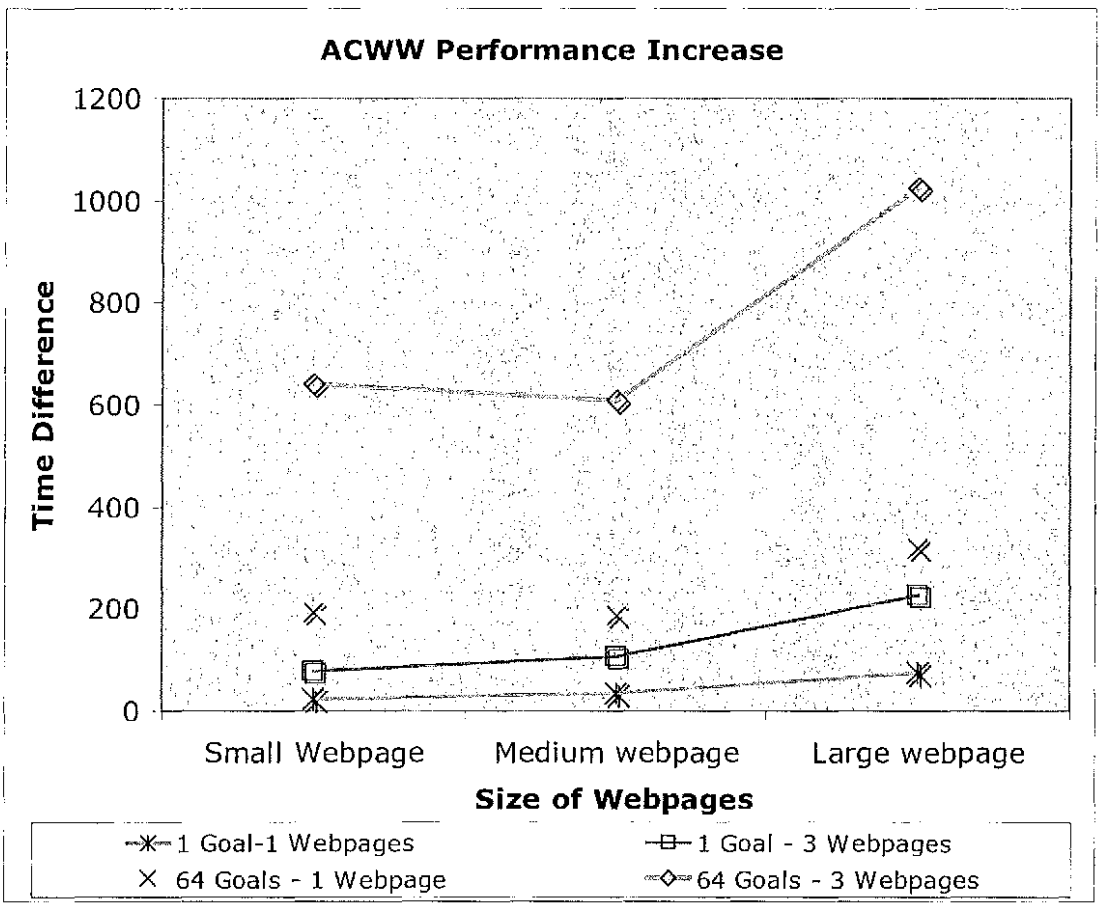


Figure 13: Time savings in relation to size of webpages

In each case (small, medium, and large webpages), the time savings grow exponentially as the website complexity grows.

This can be seen in Figure 14, Figure 15, and Figure 16. An exponential trend line was added to each graph to see how closely the performance savings matches exponential growth.

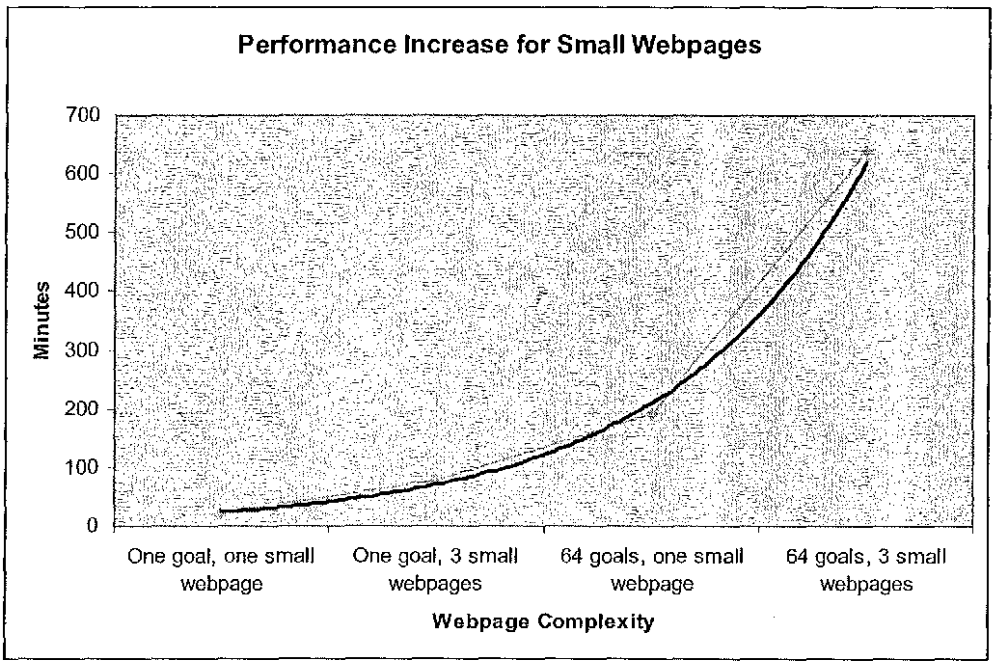


Figure 14: Performance Increase for Small Webpages

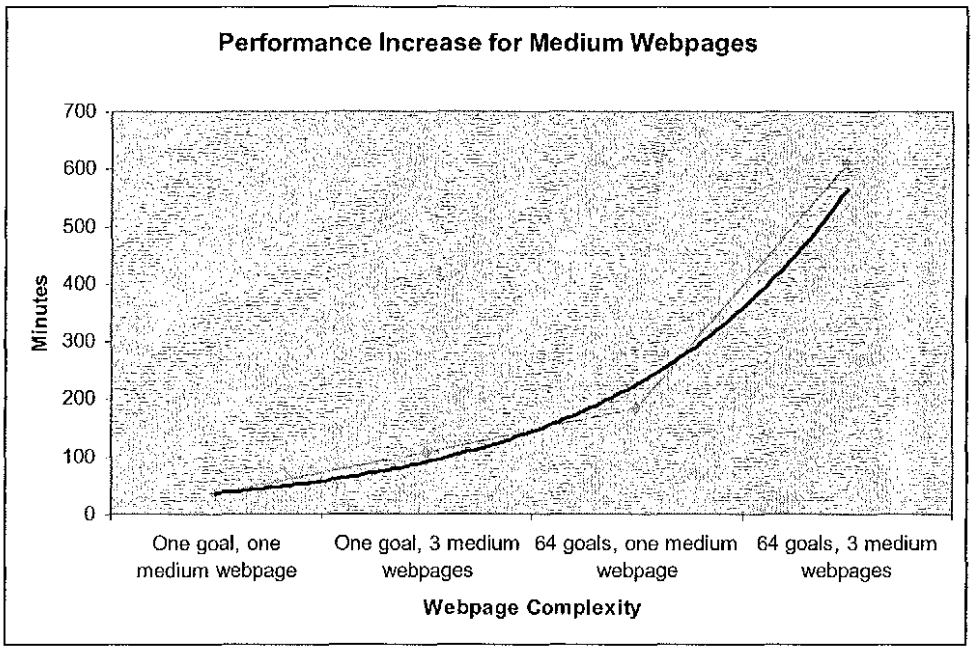


Figure 15: Performance Increase for Medium Webpages

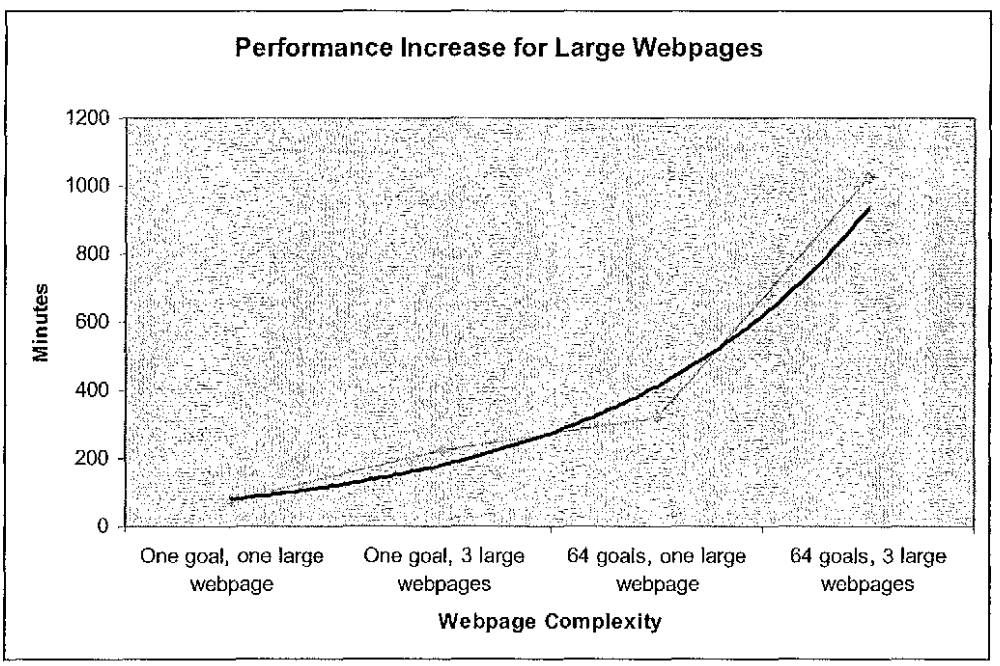


Figure 16: Performance increase for Large Webpages

6.3. Validation

Several results from previous analysis under the old method were used to compare the results from ACWW. ACWW was able to replicate the same results, only much faster and without human error.

In several instances during testing, the results from ACWW did not match the results from the manual method. The first thought was that ACWW is incorrect. When the data was reanalyzed, we found that the results from the manual method were off. Human error easily creeps into the results of the manual method. ACWW reduces human error by removing as many touch points as currently possible. Because the more manual process is tedious and repetitive, human errors are quite common. Because of this ACWW provides more consistent results.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

7.1. Conclusions

As expected, ACWW was able to duplicate the results associated with performing CWW manually. Additionally, it was able to run the same analysis 5.61 times faster, on average, compared to the old method for a single semantic space and set of parameters. It is able to run up to 100 times faster when the researcher wants to repeat the analysis with multiple semantic spaces and/or multiple parameters.

Some improvements are needed on the interface. First and foremost, the ability to backtrack and make changes to the answers already provided. This will allow the researcher to correct any mistakes made along the way without restarting. In addition, the researcher should be able to save an analysis. This will allow the researcher to revisit any set of data and perform additional analysis in without having to resubmit the data.

Since ACWW was written modularly, an additional feature should be added to take preformatted text file input so the user does not have to copy and paste the heading and links into the interface. As our understanding of how we segment webpages into visio-spatial semantic groups grows, we could replace the interface with an engine that can separate these groups automatically. The researcher would then only need to point ACWW to a specific webpage.

ACWW was built modularly so any part of it can be replaced as understanding of CW grows. It can be built upon and extended to add functionality and increase the already substantial time savings given over the current method.

APPENDIX A

AUTOMATABLE RULES FOR CWW PROBLEM-IDENTIFICATION

A.1. Classify it as a competing heading

Rule 1

If the heading is not a correct heading

And if the heading has a goal-heading cosine ≥ 0.8 times the goal-heading cosine of the correct heading or the goal-heading cosine of the correct heading that has the highest goal-heading cosine if there are two or more correct headings

And if the goal-heading cosine of the heading ≥ 0.10 (i.e., NOT weak scent)

And if the highest goal-link cosine of links nested under the heading ≥ 0.20

Then classify it as a competing heading.

Rationale: Users' attention is pulled to headings that are stronger than the correct heading, but it is meaningless to speak of "stronger than" when the higher goal-heading cosine is a weak-scent heading "in the noise." Requiring the competing heading to have a goal-heading cosine ≥ 0.8 times the goal-heading

cosine of the correct heading is consistent with the way we compute competing links (i.e., as links with goal-link cosines ≥ 0.8 times the goal-link cosine of the correct heading).

Rule 2

If the heading is not a correct heading

And if the highest-cosine link nested under that heading ≥ 0.30 (i.e., strong-scent link)

Then classify it as a competing heading.

Rationale: Sometimes users are drawn to a particular heading by a strong information scent for a specific link(s) nested under the heading. Even if the strong-scent link does not work, the user will then search for other links similar to the strong-scent link under the same heading. For example, a person might first think "Chemistry" and then look for the heading where they would find the "Chemistry" link, i.e., "Physical Science & Technology." Even if "Chemistry" turns out to not work, the user will think, "I must be close" and continue to search for other links with sufficient scent under the same goal.

A.2. Classify it as a competing heading competing link

Rule 1

If the link is nested under a competing heading

And if the goal-link cosine of the link ≥ 0.8 times the highest goal-link cosine of all the links nested under the competing heading

And if the goal-link cosine of the link ≥0.10

And if the goal-link cosine of the link is ranked no lower than fourth place when the goal-link cosines of links under the same heading are ranked in descending order, or if the goal-link cosine ≥ 0.30 (i.e., a strong-scent link)

Then classify it as a competing heading competing link.

Rationale: If the user's attention has been drawn to a competing heading, the user is apt to click links under that heading in order of decreasing information scent and then give up after clicking all the links that are NOT weak-scent or after clicking several of the high scent links under that heading.

Rule 2

If the link is nested under a competing heading

And if the goal-link cosine of the link ≥ 0.20

And if there is no more than one link under the same

heading with a higher goal-link cosine

Then classify it as a competing heading competing link.

Rationale: There are subregions where the highest-ranking link has such strong information scent that no other links under the same heading are ≥ 0.8 times the highest-ranking link. Nevertheless, users who focus on a heading and click the link with the highest goal-link cosine in that subregion, are likely to click at least one more link in that same subregion if they see one with fairly strong information scent (operationally defined as a goal-link cosine ≥ 0.20).

A.3. Classify it as a correct heading competing link

If the link is nested under a correct heading

And if the goal-link cosine of the link ≥ 0.8 times the goal-link cosine of the correct link

And if the goal-link cosine of the link ≥ 0.10

And if the goal-link cosine of the link is ranked no lower than fourth place when the goal-link cosines of links under the same heading are ranked in descending order, or if the goal-link cosine ≥ 0.30 (i.e., a strong-scent link)

Then classify it as a correct heading competing link.

Rationale: If the user's attention has been drawn to the correct heading, the user is apt to click links under that heading in order of decreasing information scent and then give up after clicking all the links that are NOT weak-scent or after clicking several of the high scent links under that heading.

A.4. Classify it as a weak-scent correct link

If the link has a goal-link cosine < 0.10

And if the link is a correct link

And if there are no correct links with a goal-link cosine ≥0.10

Then classify it as a weak-scent correct link.

Rationale: In this case competing links the weak information scent on the correct link makes the link an unlikely target of action, whether or not there is competition from other, higher-scent links.

A.5. Classify it as an unfamiliar correct link

If the text of a correct link is unfamiliar (i.e., if it has only one word and the word has a term vector length ≤0.55 or if the text of a correct link contains two or more words with a term vector length <0.80)

And if there are no correct links that are not unfamiliar

Then classify it as an unfamiliar correct link

Rationale: Unfamiliar links tend to be ignored by users, because users do not comprehend the meaning of the link, and because the unfamiliarity reduces the information scent, even if the goal-link cosine is high.

APPENDIX B

INSTALLATION INSTRUCTIONS

B.1. Programs Needed

For the back-end JAVA application to run properly, additional files need to be installed on the server. The MYSQL java connector needs to be installed in the same directory as the class file. This allows the java application to connect to the MYSQL database. The current version can be found at http://dev.mysql.com/downloads/connector/j/3.0.html

In addition, the base64 program needs to be downloaded from http://www.fourmilab.ch/webtools/base64/. This program converts the zip file into text that we can attach to the email.

B.2. Directory Structure

For security reasons, several files are located within the home directory rather than the webpage directory. In this implementation, the home directory is /usr2/home/brownr while the webpage directory is /usr2/home/brownr/public_html.

The home directory should include:

(1) Acww-elaborate.cgi

(2) Acww-one2many.cgi

(3) Acww-termvectors.cgi

(4) Base64

(5) Diptiscripts.pm

(6) Email-rep.sh

The web directory should include:

(1) ACWW.php

(2) AutoCWW.html

(3) Backend.class

(4) Finished.html

(5) Step1.html

(6) Step2.html

(7) Step2b_1button.html

(8) Step2b_2button.html

(9) Step2b_body.html

(10) Step3_footer.html

(11) Step4.html

(12) Step4_footer.html

(13) Step4_header.html

(14) Step5.html

(15) Step6.html

In the webpage directory, a subdirectory called 'Images' is needed. This sub-directory should include these files:

(1) Step1a.jpg

(2) Step1u.jpg

(3) Step2a.jpg

(4) Step2u.jpg

(5) Step3a.jpg

(6) Step3u.jpg

(7) Step4a.jpg

(8) Step4u.jpg

Finally, the MYSQL java connector should be decompressed within the webpage directory

APPENDIX C

SOURCE CODE

C.1. mail-rep.sh

```
#!/bin/sh

filepath=$1
recipient=$2
subject=$3
sender=$4

`cd $filepath;/usr/bin/zip -q results.zip *.xls`

cat <<! | /usr/sbin/sendmail -t -n
MIME-Version: 1.0
From: "$sender"
To: "$recipient"
Subject: "$subject"
Content-Type: multipart/mixed; boundary="_boundarystring"
```

```
This is a multi-part message in MIME format.

--_boundarystring

Content-Transfer-Encoding: Base64

Content-Type: application/zip

Content-Disposition: attachment; filename=results.zip


`cd $filepath;/usr2/home/brownr/base64 -e results.zip`


--_boundarystring--

!
```

C.2. ACWW.php

```
<?php

if (!session_id())
{
     session_start();
}

if (!session_is_registered("current_page"))//Used to
determine which page we are on
{
     session_register("current_page");
     $_SESSION['current_page']=1;
}

if (!session_is_registered("goal_count"))//Used to keep
track of how many goals the user has entered.
{
     session_register("goal_count");
```

```
     $_SESSION['goal_count']=0;
}

if (!session_is_registered("goals"))//Matrix to store Goal
data
{
     session_register("goals");
}

if (!session_is_registered("webpage"))//Matrix to store
Webpage data
{
     session_register("webpage");
}

if (!session_is_registered("web_count"))//Used to keep
track of how many webpages the user has entered
{
     session_register("web_count");
     $_SESSION['web_count']=0;
}

if (!session_is_registered("heading_count"))//Used to keep
track of how many headings the user has entered.
{
     session_register("heading_count");
     $_SESSION['heading_count']=1;
}

if (!session_is_registered("current_link_count"))//Keeps
track of which link set has been displayed
{
     session_register("current_link_count");
     $_SESSION['current_link_count']=0;
}

if(!session_is_registered("matrix_count"))//keeps track of
which index of the matrix we are current at
{
     session_register("matrix_count");
     $_SESSION['matrix_count']=0;
}

if(!session_is_registered("matrix")) //This array is a
matrix of which webpages are to be analyzed with which
goals.
{//It also contains the correct links
```

```
     session_register("matrix");
}

if (!session_is_registered("options"))//Holds options that
each Webpage/Goal will be ran against
{
     session_register("options");
}

if (!session_is_registered("options_cnt"))//Keeps track of
which option number we are on
{
     session_register("options_cnt");
     $_SESSION['options_cnt']=0;
}

if(!session_is_registered("linkage_count"))
{
     session_register("linkage_count");
     $_SESSION['linkage_count']=0;
}

if (!session_is_registered("email_addy"))
{
     session_register("email_addy");
}

if (!session_is_registered("srand"))
{
     session_register("srand");
     $_SESSION['srand'] = srand((double) microtime() *
1000000);
}

//We start by displaying the first page
if($_SESSION['current_page']==1 &&
$_SESSION['goal_count']==0)
{
     readfile("Step1.html");
     $_SESSION['goal_count']++;
}
//The user clicked on 'Add another' or Next Step.
//We save the input into global variable and redisplay page
or move to the next page.
else if ($_SESSION['current_page']==1 &&
$_SESSION['goal_count']>=0)
{
```

```
      //Save data into goals[]
      $_SESSION['goals'][$_SESSION['goal_count']]['label']=$
_POST['label'];
      $_SESSION['goals'][$_SESSION['goal_count']]['filename'
]=$_POST['filename'];
      $_SESSION['goals'][$_SESSION['goal_count']]['goalstate
ment']=$_POST['goalstatement'];
      //Move to Step2
      if($_POST['next']=="Next Step")
      {
            readfile("Step2a.html");
            $_SESSION['current_page']++;
      }
      //Redisplay Step 1
      else if($_POST['another']=="Enter Another Goal
Statment")
      {
            readfile("Step1.html");
            $_SESSION['goal_count']++;
      }

}
//The user just submitted data in step2.
else if($_SESSION['current_page']==2)
{
      //We break up the data that is seperated by a blank
line and save.
      //The data is the webpage label followed by the
various headings.
      $_SESSION['webpage'][$_SESSION['web_count']]['label']=
$_POST['label'];
      $headings = preg_split ("/\s\s+/",
$_POST['headings'],-1, PREG_SPLIT_NO_EMPTY);
      $h_cnt=0;
      foreach ($headings as $h)
      {

      $_SESSION['webpage'][$_SESSION['web_count']]['headings
'][$h_cnt]['name']=$h;
            $h_cnt++;
      }
      $_SESSION['current_page']++;
      $_SESSION['current_link_count']=0;
      //Display the current page header, data, then main
body.  This was done to allow easy changes to the pages.
      readfile("Step2b_header.html");
```

```
     print
(trim($_SESSION['webpage'][$_SESSION['web_count']]['heading
s'][$_SESSION['current_link_count']]['name']));
     readfile("Step2b_body.html");
     //We display certain buttons depending on if we have
reached the end of the heading matrix.
     if ($_SESSION['current_link_count'] <
count($_SESSION['webpage'][$_SESSION['current_link_count']]
['headings'])-1)
     {
          readfile("Step2b_1button.html");
     }
     else
     {
          readfile("Step2b_2button.html");
     }
}
//The user just submitted the headings and we now need to
associate the links the user just submitted.
else if($_SESSION['current_page']==3)
{
     //The links were submitted, so we break them up based
on the white space inbetween.
     $_SESSION['webpage'][$_SESSION['web_count']]['headings
'][$_SESSION['current_link_count']]['links'] = preg_split
("/\s\s+/", $_POST['links'],-1, PREG_SPLIT_NO_EMPTY);
     if ($_POST['another']=="Enter the Headings and Links
of Another Webpage")
     {
          readfile("Step2a.html");
          $_SESSION['current_page']=2;
          $_SESSION['current_link_count']=0;
          $_SESSION['web_count']++;
     }
     //If the user Clicked on next step, then we have to
decide which webpages to analyze with the goals.
     else if ($_POST['next']=="Next Step")
     {
          $_SESSION['current_page']=4;
          readfile("Step3_header.html");
          print "<tr><td></td>";
          $matrix_cnt=0;
          //This populates the first row with the webpage
labels
          for ($x=0; $x<=$_SESSION['web_count']; $x++)
          {
               print "<td><center><b>";
```

```
                print $_SESSION['webpage'][$x]['label'];
                print "</b></center></td>";
            }
            print "</tr>";
            //Next, we have to do the goals label then check
boxes.
            for ($table_goal=1;
$table_goal<=$_SESSION['goal_count']; $table_goal++)
            {
                //The next line populates the first column
of each row.
                print "<tr><td width=\"25%\"><center><b>" .
$_SESSION['goals'][$table_goal]['label'] .
"</b></center></td>";
                //The for loop populates all the rest of the
columns with a check box.  The check box name is built of
                //the goal # and webpage #
                for ($webpages=0; $webpages <=
$_SESSION['web_count']; $webpages++)
                {
                    print "<td><center>Include WebPage
<br>for analysis:<br>";
                    print "<input name=\"" . $matrix_cnt .
"\" type=\"checkbox\" value=\"" . $matrix_cnt . "\"
checked>";
                    print "<input name=\"" . $matrix_cnt .
"_goal\" type=\"hidden\" value=\"" . $table_goal . "\">";
                    print "<input name=\"" . $matrix_cnt .
"_webpage\" type=\"hidden\" value=\"" . $webpages . "\">";
                    print "<center></td>";

    $_SESSION['matrix'][$matrix_cnt]['analyze']=0;
                    $matrix_cnt++;
                }
                print "</tr>";
            }
            readfile("Step3_footer.html");
            $_SESSION['matrix_count']=$matrix_cnt;
    }
    else //if($_SESSION['current_link_count']-1 <
count($_SESSION['webpage'][$_SESSION['web_count']]['heading
s']))
    {
        //The user still needs to input more links for
the various headings.  Here, we just redisplay the last
page
```

```
          //with the next heading listed.  We run through
the entire else if until all the headings have links.
          $_SESSION['current_link_count']++;
          readfile("Step2b_header.html");
          print
(trim($_SESSION['webpage'][$_SESSION['web_count']]['heading
s'][$_SESSION['current_link_count']]['name']));
          readfile("Step2b_body.html");
          if ($_SESSION['current_link_count'] <
count($_SESSION['webpage'][$_SESSION['web_count']]['heading
s'])-1)
          {
               readfile("Step2b_1button.html");
          }
          else
          {
               readfile("Step2b_2button.html");
          }
     }
}
//The user just finished selecting the webpages/goals to be
analyized.  Next, we need to discern which
//is the correct heading/link for each goal statement.
else if ($_SESSION['current_page']==4)
{
//Here, we are saving the webpage/goals selected into a
matrix.
     if($_SESSION['linkage_count']==0)
     {
          for ($x=0;$x<$_SESSION['matrix_count'];$x++)
          {
               if ($_POST[$x]==$x)
               {
                    $_SESSION['matrix'][$x]['analyze']=1;

     $_SESSION['matrix'][$x]['goal']=$_POST[$x . "_goal"];

     $_SESSION['matrix'][$x]['webpage']=$_POST[$x .
"_webpage"];
               }
          }
     }
     if ($_SESSION['linkage_count']!=0)
     { //Save data from previous page
          $current_webpage =
$_SESSION['matrix'][$_SESSION['linkage_count']-
1]['webpage'];
```

```
          $heading_cnt=0;
          $link_cnt=0;
          $correct_cnt=0;
          foreach
($_SESSION['webpage'][$current_webpage]['headings'] as
$headings)
          {
               $link_cnt=0;
               foreach
($_SESSION['webpage'][$current_webpage]['headings'][$headin
g_cnt]['links'] as $link)
               {
                    if ($_POST['H' . $heading_cnt . 'L' .
$link_cnt ] == 'H' . $heading_cnt . 'L' . $link_cnt)
                    {

     $_SESSION['matrix'][$_SESSION['linkage_count']-
1]['correct'][$correct_cnt]['heading']=$heading_cnt;

     $_SESSION['matrix'][$_SESSION['linkage_count']-
1]['correct'][$correct_cnt]['link']=$link_cnt;
                              $correct_cnt++;

                    }
                    $link_cnt++;
               }
               $heading_cnt++;
          }
     }

     //Next, we determine if we have cycled through all the
matrix.  If we have not, the user still has
     //some associations left.
     if
($_SESSION['linkage_count']<$_SESSION['matrix_count'])
     {
          //We need to find the next association in the
matrix
          while
($_SESSION['matrix'][$_SESSION['linkage_count']]['analyze']
!=1 &&
$_SESSION['linkage_count']<$_SESSION['matrix_count'])
          {
               $_SESSION['linkage_count']++;
          }
          //then we display it...
```

```
          if
($_SESSION['linkage_count']<$_SESSION['matrix_count'])
          {
               $current_webpage =
$_SESSION['matrix'][$_SESSION['linkage_count']]['webpage'];
               readfile("Step4_header.html");
               print "<h3><center><b>For this goal
Statement:</b></center></h3>";
               print "<center>" .
$_SESSION['goals'][$_SESSION['matrix'][$_SESSION['linkage_c
ount']]['goal']]['goalstatement'] . "</center>";
               print "<h3><center><b>Please choose the
correct heading(s) and link(s) to fulfill this goal
statement.</b></center></h3><br>";
               print
$_SESSION['webpage'][$current_webpage]['label'] .
"<br><br>";

               $heading_cnt=0;
               $link_cnt=0;
               foreach
($_SESSION['webpage'][$current_webpage]['headings'] as
$headings)
               {
                    //This displays a check box for the
heading
                    print "<b><input type=checkbox
name=\"h" . $heading_cnt . "\" value=\"h" . $heading_cnt
                         . "\"> " . $headings['name']
. "</b><br>";
                    $link_cnt=0;
                    foreach
($_SESSION['webpage'][$current_webpage]['headings'][$headin
g_cnt]['links'] as $link)
                    {
                         //This displays the link that is
associated with the heading.
                         print
"    <input type=checkbox name=\"H" .
$heading_cnt . "L" . $link_cnt
                              . "\" value=\"H" .
$heading_cnt . "L". $link_cnt . "\"> " . $link .
"<br>";
                         $link_cnt++;
                    }
                    $heading_cnt++;
               }
```

```
                $_SESSION['linkage_count']++;
                readfile("Step4_footer.html");
            }
            //The user has chosen all of the correct
heading/links and we can move to step 5.  This just
displays the page.
            else if
($_SESSION['linkage_count']>=$_SESSION['matrix_count'])
            {
                readfile("Step5.html");
                $_SESSION['current_page']=5;
            }
        }
        //The user has chosen all of the correct heading/links
and we can move to step 5.  This just displays the page.
        else if
($_SESSION['linkage_count']>=$_SESSION['matrix_count'])
        {
            readfile("Step5.html");
            $_SESSION['current_page']=5;
        }
}
else if ($_SESSION['current_page']==5)
{
        $_SESSION['options'][$_SESSION['options_cnt']]['space'
]=$_POST['space'];
        $_SESSION['options'][$_SESSION['options_cnt']]['link_e
laboration']=$_POST['link_elaboration'];
        $_SESSION['options'][$_SESSION['options_cnt']]['link_f
requency']=$_POST['link_frequency'];
        $_SESSION['options'][$_SESSION['options_cnt']]['link_c
osine']=$_POST['link_cosine'];
        $_SESSION['options'][$_SESSION['options_cnt']]['headin
g_elaboration']=$_POST['heading_elaboration'];
        $_SESSION['options'][$_SESSION['options_cnt']]['headin
g_frequency']=$_POST['heading_frequency'];
        $_SESSION['options'][$_SESSION['options_cnt']]['headin
g_cosine']=$_POST['heading_cosine'];
        $_SESSION['options'][$_SESSION['options_cnt']]['base']
=$_POST['base'];
        $_SESSION['options'][$_SESSION['options_cnt']]['unfami
liar']=$_POST['unfamiliar'];
        $_SESSION['options'][$_SESSION['options_cnt']]['weaksc
ent']=$_POST['weakscent'];
        $_SESSION['options'][$_SESSION['options_cnt']]['clcomp
etingheadings']=$_POST['clcompetingheadings'];
```

```
     $_SESSION['options'][$_SESSION['options_cnt']]['clcorr
ectheadings']  =$_POST['clcorrectheadings'];
     $_SESSION['options'][$_SESSION['options_cnt']]['compet
ingheadings']=$_POST['competingheadings'];

     if ($_POST['next']=="Next Step")
     {
          readfile("Step6.html");
          $_SESSION['current_page']=6;
          $_SESSION['options_cnt']++;
     }
     else //User wants to enter another set of options
     {
          readfile("Step5.html");
          $_SESSION['options_cnt']++;
     }
}
else if ($_SESSION['current_page']==6)
{
     $_SESSION['email_addy']=$_POST['email_addy'];
     // Connecting, selecting database
     $mylink = mysql_connect('localhost', 'brownr',
'password') or die('Could not connect: ' . mysql_error());
//   $mylink = mysql_connect('localhost', 'richardbrown')
or die('Could not connect: ' . mysql_error());
     mysql_select_db('acww') or die('Could not select
database');
     $rand_num = rand();

     $list_of_job_ids="";

     for ($i=0; $i < $_SESSION['options_cnt']; $i++)
     {
          $query = "INSERT INTO job_options (job_space,
job_link_freq, job_link_cosine, job_head_freq,
job_head_cosine
                    , job_email, job_id, base, unfamiliar,
weakscent, competingheadings,
competinglinksundercorrectheadings,
competinglinksundercompetingheadings)
                    VALUES ('" .
$_SESSION['options'][$i]['space'] . "','";
          if
($_SESSION['options'][$i]['link_elaboration']=="no")
               $query = $query . "-2', '-2";
          else
```

```
                $query = $query .
$_SESSION['options'][$i]['link_frequency'] . "','" .
$_SESSION['options'][$i]['link_cosine'];

          if
($_SESSION['options'][$i]['heading_elaboration']=="no")
                $query = $query . "',' -2', '-2',";
          else if
($_SESSION['options'][$i]['heading_elaboration']=="minimum"
)
                $query = $query . "', '-1', '-1',";
          else
                $query = $query . "','" .
$_SESSION['options'][$i]['heading_frequency'] . "','"
                          .
$_SESSION['options'][$i]['heading_cosine'] . "',";

          $query = $query . "'" . $_SESSION['email_addy'] .
"','" . session_id() . $rand_num . $i . "','" .
$_SESSION['options'][$i]['base'] . "','"
                          .
$_SESSION['options'][$i]['unfamiliar'] . "','" .
$_SESSION['options'][$i]['weakscent'] . "','"
                          .
$_SESSION['options'][$i]['competingheadings'] . "','" .
$_SESSION['options'][$i]['clcorrectheadings'] . "','"
                          .
$_SESSION['options'][$i]['clcompetingheadings'] . "');";

          $result = mysql_query($query) or
die(mysql_error());//submit job_options data
          $job_num = mysql_insert_id();
          $list_of_job_ids = $list_of_job_ids . " " .
$job_num;

          for ($table_goal=1; $table_goal <=
$_SESSION['goal_count']; $table_goal++)
          {//saves all information concerning the goal
statment into the database
                $query = "INSERT INTO goal (goal_label,
goal_statement, goal_filename, job_num) VALUES ('"
                          .
$_SESSION['goals'][$table_goal]['label'] . "','" .
$_SESSION['goals'][$table_goal]['goalstatement'] . "','"
                          .
$_SESSION['goals'][$table_goal]['filename'] . "','" .
$job_num . "');";
```

```
                $result = mysql_query($query) or
die(mysql_error());
          }//end for goal

          for ($webpages=0; $webpages <=
$_SESSION['web_count']; $webpages++)
          {//save all information concerning the webpage
(headings/links) into the database
                $query = "INSERT INTO webpage (web_label,
job_num) VALUES ('"
                                .
$_SESSION['webpage'][$webpages]['label'] . "','" . $job_num
. "');";
                mysql_query($query) or die(mysql_error());
                $web_num = mysql_insert_id();
                $h_num=0;
                foreach
($_SESSION['webpage'][$webpages]['headings'] as $headings)
                {
                     $query = "INSERT INTO heading
(heading_label, web_num, job_num) VALUES ('"
                                    . $headings['name'] . "','" .
$web_num . "','" . $job_num . "');";
                     mysql_query($query) or
die(mysql_error());
                     $heading_num=mysql_insert_id();
                     foreach
($_SESSION['webpage'][$webpages]['headings'][$h_num]['links
'] as $link)
                     {
                          $query = "INSERT INTO link
(link_label, head_num, job_num) VALUES ('"
                                       . $link . "','" .
$heading_num . "','" . $job_num . "');";
                          mysql_query($query) or
die(mysql_error());
                     }
                     $h_num++;
                }
          }//end for webpages

          $imatrix=0;
          foreach ($_SESSION['matrix'] as $matrix)
          {
                if ($matrix['analyze']==1)
                {
```

```
                    $query = "SELECT goal_num FROM goal
where job_num='" . $job_num . "' AND goal_label='"
                                .
$_SESSION['goals'][$matrix['goal']]['label'] . "';";
                    $result = mysql_query($query) or
die(mysql_error());
                    $goal_num = mysql_result($result,0);
                    $query = "SELECT web_num FROM webpage
where job_num='" . $job_num . "' AND web_label='"
                                .
$_SESSION['webpage'][$matrix['webpage']]['label'] . "';";
                    $result = mysql_query($query) or
die(mysql_error());
                    $web_num = mysql_result($result,0);
                    $query = "INSERT INTO matrix (web_num,
job_num, goal_num) VALUES ('"
                                . $web_num . "','" . $job_num
. "','" . $goal_num . "');";
                    mysql_query($query) or
die(mysql_error());
                    $matrix_num = mysql_insert_id();

                    foreach
($_SESSION['matrix'][$imatrix]['correct'] as $correct)
                    {//cross reference the heading# we have
in the script with the heading# that the database has
                        $query = "SELECT heading_num FROM
heading WHERE web_num='" . $web_num . "' AND
heading_label='"
                                .
$_SESSION['webpage'][$matrix['webpage']]['headings'][
$correct['heading'] ]['name'] . "';";
                        $result = mysql_query($query);
                        $mheading_num =
mysql_result($result,0) or die(mysql_error());

                        $query = "INSERT INTO
correct_heading (matrix_num, head_num, job_num) VALUES ('"
. $matrix_num
                                . "', '" . $mheading_num
. "','" . $job_num . "');";
                        mysql_query($query) or
die(mysql_error());
                        $cheading = mysql_insert_id();

                        $ilink=0;//keeps track of current
link we are cycling through
```

```
                        foreach($_SESSION['webpage'][
$matrix['webpage'] ]['headings'][ $correct['heading']
][links'] as $flink)
                        {
                            if
($ilink==$correct['link'])//compares current link# to
stored link # that is correct
                            {//find a matching link so we
can grab the index from the database
                                $query = "SELECT
link_num FROM link WHERE head_num='" . $mheading_num . "'
AND link_label='"
                                    . $flink . "';";
                                $result =
mysql_query($query) or die(mysql_error());
                                $mlink =
mysql_result($result, 0);

                                $query = "INSERT INTO
correct_link (correct_head_num, link_num, job_num) VALUES
('" . $cheading . "','" . $mlink . "','" . $job_num .
"');";
                                mysql_query($query) or
die(mysql_error());
                            }
                            $ilink++;
                        }//end for each link
                    }//end for each correct
                }//end if analyze==1
                $imatrix++;
            }//end for matrix
        }//end for options_cnt

        set_time_limit(0);
        //$pHandle = popen("java
/Users/richardbrown/Sites/ACWW/current/thesis.thesis & >
t.txt", "r");
        exec("nohup /usr/java/jdk1.3.1_13/bin/java backend " .
$list_of_job_ids . " >/dev/null 2>&1 &");
//      exec("nohup echo \"/usr/java/jdk1.3.1_13/bin/java
backend " . $list_of_job_ids . " >/dev/null 2>&1 &\" >
/tmp/command.log");
//      exec("nohup echo \"Options_cnt: " .
$_SESSION[options_cnt] . "\" >> /tmp/command.log");
//      exec("nohup chmod a+x /tmp/command.log");
        mysql_close($mylink);
```

```
      //We are finally finished and we destroy the setting
and remove the globals.
      // Unset session data
      $_SESSION=array();
      // Clear cookie
      unset($_COOKIE[session_name()]);
      // Destroy session data
      session_destroy();

//    session_unset();
      readfile("Finished.html");
}

?>
```

C.3. backend.java

```
import java.io.*;
import java.util.*;
import java.util.Random;
import java.math.*;
import java.lang.*;
import java.sql.*;
import java.net.URLEncoder;

public class backend {
//private String url = "jdbc:mysql://" + "localhost" + "/"
+ "acww?user=root";
private String url = "jdbc:mysql://" + "localhost" + "/" +
"acww?user=brownr&password=password";
//private String url = "jdbc:mysql://" + "192.168.1.104" +
"/" + "acww?user=root";

  public backend(String[] args)
  {
    try
    {
      Thread.sleep(50);//wait_time);
    }
    catch(InterruptedException e)
    {
```

```
      System.out.println("Sleep Interrupted");
    }
    this.elaborate_headings_and_links(args);  //Performs
heading/link elaboration and save into the database
    this.perform_one_to_many(args); //Performs one2many
analysis
    this.create_report(args);
    this.cleanup(args);
    System.out.println("DONE");

  }//end public backend
//-------------------------------------------------------------
-----------------------------------------------
//          Calculate Predicted Mean Clicks then Create
Report
//-------------------------------------------------------------
-----------------------------------------------
private void cleanup (String[] args)
{
  try
  {
    try
    {
      Thread.sleep(1000000);//wait_time);
    }
    catch(InterruptedException e)
    {
      System.out.println("Sleep Interrupted");
    }
    for (int a=0; a < args.length; a++)
    {
      Class.forName("com.mysql.jdbc.Driver").newInstance();
      Connection conn = DriverManager.getConnection(url);

      String sql = "DELETE FROM correct_link WHERE
job_num=" + args[a];
      PreparedStatement ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM link WHERE job_num=" + args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM correct_heading WHERE job_num=" +
args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM heading WHERE job_num=" + args[a];
      ps = conn.prepareStatement(sql);
```

```
      ps.execute();
      sql = "DELETE FROM one_to_many_results WHERE
job_num=" + args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM matrix WHERE job_num=" + args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM webpage WHERE job_num=" + args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM goal WHERE job_num=" + args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();
      sql = "DELETE FROM job_options WHERE job_num=" +
args[a];
      ps = conn.prepareStatement(sql);
      ps.execute();

      ps.close();
      conn.close();
    }//end for
  }//end try
     catch ( java.sql.SQLException e
){e.printStackTrace();}
     catch ( java.lang.ClassNotFoundException e
){e.printStackTrace();}
     catch ( java.lang.IllegalAccessException e
){e.printStackTrace();}
     catch ( java.lang.InstantiationException e
){e.printStackTrace();}
}
//-------------------------------------------------------
-----------------------------------------
//          String conversion
//-------------------------------------------------------
-----------------------------------------
  private String fix(String arg)
  {
       char [] original = new char[arg.length()];
    char [] new_array = new char[arg.length()*5];
    original = arg.toCharArray();
    int z=0;
       for (int a=0; a < arg.length() && z <
arg.length()*5; a++)
     {
          if (original[a]=='+')
```

```
      {
        new_array[a+z] = '%';
        new_array[a+z+1] = '2';
        new_array[a+z+3] = '0';
        z+=3;
      }
         else
        new_array[a+z]=original[a];
     }
    String new_string = new String (new_array);
    return new_string;
  }

//-------------------------------------------------------------
-----------------------------------------------
//          Calculate Predicted Mean Clicks then Create
Report
//-------------------------------------------------------------
-----------------------------------------------
  private int create_report(String[] args)
  {
System.out.println("Create report");
     try
     {

Class.forName("com.mysql.jdbc.Driver").newInstance();
       Connection conn = DriverManager.getConnection(url);
       ResultSet rs = null;
       Statement stat = null;
       String sql = null;
       try
       {
          for (int a=0; a < args.length; a++)
          { //Cycle through the job_num(s) provided by the
php script

            try
            {
              //This calls the elaboration script
              String command = "mkdir /tmp/acww_job" +
args[a];
               Process ls_proc =
Runtime.getRuntime().exec(command);
              DataInputStream ls_in = new
DataInputStream(ls_proc.getInputStream());
            } catch (IOException e1)
{System.err.println(e1); System.exit(1);}
```

```
            stat = conn.createStatement();
            sql = "SELECT * FROM job_options WHERE
job_num=" + args[a];
            rs = stat.executeQuery(sql);
            rs.next();
            String job_space =  rs.getString("job_space");
            String job_email = rs.getString("job_email");
            String job_id = rs.getString("job_id");

            float job_link_freq =
rs.getFloat("job_link_freq");
            float job_link_cos =
rs.getFloat("job_link_cosine");
            float job_head_freq =
rs.getFloat("job_head_freq");
            float job_head_cos =
rs.getFloat("job_head_cosine");

            double factor_base = rs.getDouble("base");
            double factor_unfamiliar_link =
rs.getDouble("unfamiliar");
            double factor_weak_scent_link =
rs.getDouble("weakscent");
            double
factor_competing_link_under_competing_heading =
rs.getDouble("competinglinksundercompetingheadings");
            double
factor_competing_link_under_correct_heading =
rs.getDouble("competinglinksundercorrectheadings");
            double factor_competing_heading =
rs.getDouble("competingheadings");

            //Grab the webpage and goal to analyze together
            sql = "SELECT * FROM matrix WHERE job_num=" +
args[a];
            rs = stat.executeQuery(sql);
            Vector matrix_num_vector = new Vector();
            Vector matrix_web_num_vector = new Vector();
            Vector matrix_goal_num_vector = new Vector();
            while(rs.next())
            {

matrix_num_vector.add(rs.getString("matrix_num"));

matrix_web_num_vector.add(rs.getString("web_num"));
```

```
matrix_goal_num_vector.add(rs.getString("goal_num"));
            }
            int matrix_count = matrix_num_vector.size();
            String matrix_num[] = new String[matrix_count];
            String matrix_web_num[] = new
String[matrix_count];
            String matrix_goal_num[] = new
String[matrix_count];
            matrix_num_vector.copyInto(matrix_num);
            matrix_web_num_vector.copyInto(matrix_web_num);

matrix_goal_num_vector.copyInto(matrix_goal_num);


            for (int b=0; b < matrix_count; b++)
            {
              sql = "SELECT * FROM webpage WHERE web_num="
+ matrix_web_num[b];
              rs = stat.executeQuery(sql);
              rs.next();
              String web_label = rs.getString("web_label");

              sql = "SELECT * FROM one_to_many_results
WHERE matrix_num=" + matrix_num[b];
              rs = stat.executeQuery(sql);
              Vector result_num_vector = new Vector();
              Vector original_label_vector = new Vector();
              Vector elaborated_label_vector = new
Vector();
              Vector cosine_vector = new Vector();
              Vector heading_or_link_vector = new Vector();
              Vector specific_heading_vector = new
Vector();
              Vector correct_vector = new Vector();
              Vector result_heading_num_vector = new
Vector();
              while(rs.next())
              {

result_num_vector.add(rs.getString("result_num"));

original_label_vector.add(rs.getString("original_label"));

elaborated_label_vector.add(rs.getString("elaborated_label"
));
                  cosine_vector.add(rs.getString("cosine"));
```

```
heading_or_link_vector.add(rs.getString("heading_or_link"))
;

specific_heading_vector.add(rs.getString("specific_heading"
));

correct_vector.add(rs.getString("correct"));

result_heading_num_vector.add(rs.getString("heading_num"));
                }
                int result_count = result_num_vector.size();
                String result_num [] = new
String[result_count];
                String original_label [] = new
String[result_count];
                String elaborated_label [] = new
String[result_count];
                String cosine [] = new String [result_count];
                String heading_or_link [] = new
String[result_count];
                String specific_heading [] = new
String[result_count];
                String correct [] = new String[result_count];
                String classification [][] = new
String[result_count][5];
                String used [] = new String[result_count];
                double term_vectors[] = new double
[result_count];
                //classification[result_count][0]= weak-scent
correct link
                //classification[result_count][1]= unfamiliar
correct link
                //classification[result_count][2] = competing
link under competing heading
                //classification[result_count][3]= competing
link under correct heading
                //classification[result_count][4]= competing
heading
                String result_heading_num [] = new String
[result_count];
                result_num_vector.copyInto(result_num);

original_label_vector.copyInto(original_label);

elaborated_label_vector.copyInto(elaborated_label);
                cosine_vector.copyInto(cosine);
```

```
heading_or_link_vector.copyInto(heading_or_link);

specific_heading_vector.copyInto(specific_heading);
              correct_vector.copyInto(correct);

result_heading_num_vector.copyInto(result_heading_num);

              double largest_correct_heading_cosine=-1.0;
//cosine of the largest correct heading
              double largest_correct_link_cosine=-1.0;
              int num_of_correct_headings=0;
              int num_of_correct_links=0;
              int num_of_headings=0;
              int num_of_links=0;
              boolean
at_least_one_correct_link_is_GTE_point_one = false;
              boolean found_familiar_link=false;
              int weak_scent_correct_link=0;
              int unfamiliar_correct_link=0;
              int competing_heading=0;
              int competing_link_under_competing_heading=0;
              int competing_link_under_correct_heading=0;
              for (int c=0; c < result_count; c++)
              {
                classification[c][0] = " ";
                classification[c][1] = " ";
                classification[c][2] = " ";
                classification[c][3] = " ";
                classification[c][4] = " ";

                used[c] = "Not Used";
              }
              //Find largest correct heading/link and the
number of correct headings
              for (int c=0; c < result_count; c++)
              {
                if (heading_or_link[c].equals("Heading"))
                  num_of_headings++;
                if (heading_or_link[c].equals("Link"))
                  num_of_links++;
                if (correct[c].equals("Yes") &&
heading_or_link[c].equals("Heading"))
                {
                  Double hcs = new Double(cosine[c]);
                  num_of_correct_headings++;
```

```
                    if ( hcs.doubleValue() >
largest_correct_heading_cosine)
                      largest_correct_heading_cosine =
hcs.doubleValue();
                  }
                  else if (correct[c].equals("Yes") &&
heading_or_link[c].equals("Link"))
                  {
                    Double hcs = new Double(cosine[c]);
                    num_of_correct_links++;
                    if ( hcs.doubleValue() >
largest_correct_link_cosine)
                      largest_correct_link_cosine =
hcs.doubleValue();
                  }
                  if (correct[c].equals("Yes") &&
heading_or_link[c].equals("Link"))
                  {
                    Double hcs = new Double(cosine[c]);
                    if (hcs.doubleValue() >= 0.10)

at_least_one_correct_link_is_GTE_point_one = true;

//                    if (
(this.countWords(original_label[c])==1 && hcs.doubleValue()
> 0.55) || (this.countWords(original_label[c])>1 &&
hcs.doubleValue() > 0.80) )
//                      found_familiar_link=true;
                  }

                }//end for int c

                //Classify competing heading
                for (int c=0; c < result_count; c++)
                {
                  //Look at incorrect headings/Links
                  if (correct[c].equals("No") &&
heading_or_link[c].equals("Heading"))  //We don't need to
examine correct headings or links
                  {
                    Double hcs = new
Double(cosine[c]);//cosine of current heading
                    boolean found_competing_heading = false;
                    for (int d=0; d < result_count &&
found_competing_heading==false; d++)
                    {
```

```
                    if (
result_heading_num[d].equals(result_heading_num[c]) &&
heading_or_link[d].equals("Link") ) //look @ links under
current heading
                    {
                      Double cs = new
Double(cosine[d]);//cosine of current link
                      if (cs.doubleValue() >= 0.30)
//fulfills rule 4
                      {
                        classification[c][4] = "Competing
Heading";//found a competing heading.  Rule 4
                        competing_heading++;
                        found_competing_heading=true;
                      }
                      else if (cs.doubleValue() >= (.8 *
largest_correct_heading_cosine) && hcs.doubleValue() >=
0.10 && cs.doubleValue() > 0.20) //fulfills rule 5
                      {
                        classification[c][4] = "Competing
Heading";//found a competing heading.         Rule 3
                        competing_heading++;
                        found_competing_heading=true;
                      }
                    }//end if
                  }//end for int d
                }//end if
                else if (correct[c].equals("Yes") &&
heading_or_link[c].equals("Heading"))
                {   //Find Correct Heading Competing Link
                  Double hcs = new
Double(cosine[c]);//cosine of current heading
                  for (int d=0; d < result_count; d++)
//cycle looking for links underneath correct heading
                  {
                    if (
result_heading_num[d].equals(result_heading_num[c]) &&
heading_or_link[d].equals("Link") &&
correct[d].equals("No")) //look @ links under current
heading
                    {
                      Double cs = new
Double(cosine[d]);//cosine of current link
                      if (cs.doubleValue() >= (0.8 *
largest_correct_link_cosine) && cs.doubleValue() >= 0.10)
                      {
                       if (cs.doubleValue() >= 0.30)
```

```
                        {
                         classification[d][3] = "Competing
Link under Correct Heading"; //Rule 5

competing_link_under_correct_heading++;
                        }
                        else
                        {
                          int rank=0;
                          for (int e=0; e < result_count;
e++)
                          {
                            if (
result_heading_num[e].equals(result_heading_num[c]) &&
heading_or_link[e].equals("Link") &&
correct[e].equals("No")) //look @ links under current
heading
                            {
                              Double es = new
Double(cosine[d]);//cosine of current link
                              if (es.doubleValue() >
cs.doubleValue() && es.doubleValue() != cs.doubleValue())
                                rank++;
                            }
                          }//end for e
                          if (rank < 5  &&
correct[d].equals("No"))
                          {
                            classification[d][3] =
"Competing Link under correct heading"; //Rule 5

competing_link_under_correct_heading++;
                          }
                        }//end else
                      }//end if
                    }//end if
                  }//end for d
                }//end else if
              }//end for int c

              //Classify weak-scent correct link &
Unfamiliar Correct Link
              if
(at_least_one_correct_link_is_GTE_point_one == false)
              {
                for (int c=0; c < result_count; c++)
                {
```

```
                    Double hcs = new
Double(cosine[c]);//cosine of current item
                    if (correct[c].equals("Yes") &&
heading_or_link[c].equals("Link") && hcs.doubleValue() <
0.10 )
                    {
                      classification[c][0] = "Weak-Scent
Correct Link";     //Rule 1
                      weak_scent_correct_link++;
                    }
                  }//end for int c
                } //end if
at_least_one_correct_link_is_GTE_point_one

//                if (found_familiar_link==false)
//                {
                  for (int c=0; c < result_count; c++)
                  {
                      Double term_vector=new Double(0.0);
                      try
                      {
                        String eL =
URLEncoder.encode(original_label[c]);
                        String command =
"/usr2/home/brownr/acww-termVectors.cgi Space=" + job_space
+ " Links=" + eL;
                         Process ls_proc =
Runtime.getRuntime().exec(command);
                           DataInputStream ls_in = new
DataInputStream(ls_proc.getInputStream());
                           try
                        {
                          String ls_str;
                            while ((ls_str =
ls_in.readLine()) != null)
                           {
                             if
(ls_str.startsWith("LSApseudodoc.pm"))
                               term_vector = new
Double(99999);
                           else
                             term_vector = new
Double(ls_str);
                                 }
                        } catch (IOException e1)
{System.err.println(e1); System.exit(1);}
```

```
                    }catch (IOException e1)
{System.err.println(e1); System.exit(1);}
                    term_vectors[c] =
term_vector.doubleValue();
System.out.println("Term Vector: " + term_vectors[c]);
//                   Double hcs = new
Double(cosine[c]);//cosine of current item
                   if ( heading_or_link[c].equals("Link") &&
correct[c].equals("Yes")  )
                   {

                    if (
(this.countWords(original_label[c])==1 &&
term_vector.doubleValue() <= 0.55) ||
(this.countWords(original_label[c])>1 &&
term_vector.doubleValue() < 0.80) )
                    {
                     classification[c][1] = "Unfamiliar
Correct Link"; //Rule 2
                     unfamiliar_correct_link++;
                     }//end if
                    }//end if
                   }//end for
//                }//end if (found_familiar_link==false)

               for (int c=0; c < result_count; c++)
               {
                 //Look at competing headings
                 if (classification[c][4].equals("Competing
Heading") && heading_or_link[c].equals("Heading"))
                   {
                     Double hcs = new
Double(cosine[c]);//cosine of current heading
                     for (int d=0; d < result_count; d++)
                     {
                       if (
result_heading_num[d].equals(result_heading_num[c]) &&
heading_or_link[d].equals("Link") &&
correct[d].equals("No")) //look @ links under current
heading
                       {
                         Double cs = new
Double(cosine[d]);//cosine of current link
                        if (cs.doubleValue() >= 0.10)
                        {
                          int rank=0;
```

```
                        double max_cosine_under_heading = -
1.0;
                        for (int e=0; e < result_count;
e++)
                        {
                          Double es = new
Double(cosine[e]);//cosine of current link
                          if (es.doubleValue() >
cs.doubleValue() && es.doubleValue() != cs.doubleValue())
                            rank++;
                          if (es.doubleValue() >
max_cosine_under_heading)

max_cosine_under_heading=es.doubleValue();
                        }
                        if ( (rank <= 4 ||
cs.doubleValue()>= 0.30) && cs.doubleValue() >= (.08 *
max_cosine_under_heading) )
                        {
                          classification[d][2] = "Competing
Link under Competing Heading";  //Rule 6a

competing_link_under_competing_heading++;
                        }

                       }

                     else if (cs.doubleValue() >= 0.20  )//
&& cs.doubleValue() >=  largest_correct_link_cosine)
                     {
                      int rank=0;
                      for (int e=0; e < result_count; e++)
                      {
                        Double es = new
Double(cosine[e]);//cosine of current link
                        if (es.doubleValue() >
cs.doubleValue() && es.doubleValue() != cs.doubleValue())
                          rank++;
                      }
                      if (rank <= 1)
                      {
                        classification[d][2] = "Competing
Link under Competing Heading";  //Rule 6b

competing_link_under_competing_heading++;
                      }
```

```
                    }
                   }//end if
                 }//end for int d
               }//end if
             }//end for c

             double PMC=factor_base; //Predicted Mean
Click
             boolean liu = false;
             boolean wscl = false;
             if (unfamiliar_correct_link>0)
             {
               PMC += factor_unfamiliar_link;
               liu = true;
             }
             if( weak_scent_correct_link > 0)
             {
               PMC += factor_weak_scent_link;
               wscl = true;
             }

             PMC = PMC
               +
factor_competing_link_under_competing_heading *
competing_link_under_competing_heading
               +
factor_competing_link_under_correct_heading *
competing_link_under_correct_heading
               + factor_competing_heading *
competing_heading
               ;



             String SPMC = "Predicted Mean Clicks = " +
factor_base + " + " + factor_unfamiliar_link + " (Link is
unfamiliar: " + liu + ")"
                 + " + " + factor_weak_scent_link + "
(Link has a weak-scent: " + wscl + ") "
                 + " + " +
factor_competing_link_under_competing_heading + " * "  +
competing_link_under_competing_heading + " (Number of
competing links nested under competing headings)"
                 + " + " +
factor_competing_link_under_correct_heading + " * " +
competing_link_under_correct_heading + "(Number of
competing links nested under the correct headings)"
```

```
                    + " + " + factor_competing_heading + " *
" + competing_heading + "(Number of competing headings) =
";

System.out.println("PMC: " + PMC);
                //-----------------------------------------
---------------------------
                //Finished calculating Predicted Mean Click.
Now we write it out to a file
                sql = "Select * FROM goal WHERE goal_num=" +
matrix_goal_num[b];
                rs = stat.executeQuery(sql);
                rs.next();
                String goal_label =
rs.getString("goal_label");
                String goal_statement =
rs.getString("goal_statement");
                String goal_filename =
rs.getString("goal_filename");
                FileOutputStream out; // declare a file
output object
                PrintStream p; // declare a print stream
object
                try
                {
                  out = null;
                  //exec("mkdir /tmp/" + job_id);
                  if (goal_filename.endsWith(".xls")==true)
                    out = new
FileOutputStream("/tmp/acww_job" + args[a] + "/" +
goal_filename);
                  else
                    out = new
FileOutputStream("/tmp/acww_job" + args[a] + "/" +
goal_filename + ".xls");
                  p = new PrintStream( out );
                  p.println ("<html xmlns:o=\"urn:schemas-
microsoft-com:office:office\"xmlns:x=\"urn:schemas-
microsoft-
com:office:excel\"xmlns=\"http://www.w3.org/TR/REC-
html40\">");
                  p.println ("<head><meta http-equiv=Content-
Type content=\"text/html; charset=macintosh\"><meta
name=ProgId content=Excel.Sheet><meta name=Generator
content=\"Microsoft Excel 10\"><link rel=File-List
href=\"Ammonitemodified_files/filelist.xml\"><link
rel=Edit-Time-Data
```

```
href=\"Ammonitemodified_files/editdata.mso\"><link rel=OLE-
Object-Data href=\"Ammonitemodified_files/oledata.mso\"><!-
-[if gte mso
9]><xml><o:DocumentProperties><o:LastAuthor>ACWW</o:LastAut
hor><o:LastSaved>2004-11-
20T22:52:02Z</o:LastSaved><o:Version>10.2625</o:Version></o
:DocumentProperties></xml><![endif]--><style><!--
td{padding-top:1px;}table{mso-displayed-decimal-
separator:\"\\.\";mso-displayed-thousand-
separator:\"\\,\";}");
                p.println ("@page{margin:1.0in .75in 1.0in
.75in;mso-header-margin:.5in;mso-footer-margin:.5in;}");
                p.println ("tr{mso-height-source:auto;}");
                p.println ("col{mso-width-source:auto;}");
                p.println ("br{mso-data-placement:same-
cell;}");
                p.println (".style0{mso-number-
format:General;text-align:general;vertical-
align:bottom;white-space:nowrap;mso-rotate:0;mso-
background-source:auto;mso-
pattern:auto;color:windowtext;font-size:10.0pt;font-
weight:400;font-style:normal;text-decoration:none;font-
family:Arial;mso-generic-font-family:auto;mso-font-
charset:0;border:none;mso-protection:locked visible;mso-
style-name:Normal;mso-style-id:0;}");
                p.println ("td{mso-style-
parent:style0;padding-top:1px;padding-right:1px;padding-
left:1px;mso-ignore:padding;color:windowtext;font-
size:10.0pt;font-weight:400;font-style:normal;text-
decoration:none;font-family:Arial;mso-generic-font-
family:auto;mso-font-charset:0;mso-number-
format:General;text-align:general;vertical-
align:bottom;border:none;mso-background-source:auto;mso-
pattern:auto;mso-protection:locked visible;white-
space:nowrap;mso-rotate:0;}");
                p.println (".xl24{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;}");
                p.println (".xl25{mso-style-
parent:style0;font-weight:700;}");
                p.println (".xl26{mso-style-
parent:style0;font-weight:700;font-family:Verdana, sans-
serif;mso-font-charset:0;}");
                p.println (".xl27{mso-style-
parent:style0;font-weight:700;font-family:Verdana, sans-
serif;mso-font-charset:0;text-align:left;border:.5pt solid
windowtext;white-space:normal;}");
```

```
                p.println (".xl28{mso-style-
parent:style0;font-weight:700;font-family:Verdana, sans-
serif;mso-font-charset:0;text-align:left;border-top:.5pt
solid windowtext;border-right:.5pt solid windowtext;border-
bottom:.5pt solid windowtext;border-left:none;white-
space:normal;}");
                p.println (".xl29{mso-style-
parent:style0;font-weight:700;text-align:left;border-
top:.5pt solid windowtext;border-right:.5pt solid
windowtext;border-bottom:.5pt solid windowtext;border-
left:none;white-space:normal;}");
                p.println (".xl30{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;background:yellow;mso-
pattern:auto none;}");
                p.println (".xl31{mso-style-
parent:style0;text-align:left;border-top:none;border-
right:.5pt solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;background:yellow;mso-
pattern:auto none;}");
                p.println (".xl32{mso-style-
parent:style0;color:black;font-family:Verdana, sans-
serif;mso-font-charset:0;text-align:left;border-
top:none;border-right:.5pt solid windowtext;border-
bottom:.5pt solid windowtext;border-
left:none;background:lime;mso-pattern:auto none;}");
                p.println (".xl33{mso-style-
parent:style0;color:black;text-align:left;border-
top:none;border-right:.5pt solid windowtext;border-
bottom:.5pt solid windowtext;border-
left:none;background:lime;mso-pattern:auto none;}");
                p.println (".xl34mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;}");
                p.println (".xl35mso-style-
parent:style0;text-align:left;border-top:none;border-
right:.5pt solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;}");
                p.println (".xl36{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
```

```
windowtext;border-left:none;background:lime;mso-
pattern:auto none;}");
                p.println (".xl137{mso-style-
parent:style0;text-align:left;border-top:none;border-
right:.5pt solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;background:lime;mso-
pattern:auto none;}");
                p.println (".xl138{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:.5pt solid
windowtext;background:yellow;mso-pattern:auto none;}");
                p.println (".xl139{mso-style-
parent:style0;color:black;font-family:Verdana, sans-
serif;mso-font-charset:0;text-align:left;border-
top:none;border-right:.5pt solid windowtext;border-
bottom:.5pt solid windowtext;border-
left:none;background:lime;mso-pattern:auto none;}");
                p.println (".xl140{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:.5pt solid windowtext;}");
                p.println (".xl141{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:.5pt solid
windowtext;background:lime;mso-pattern:auto none;}");
                p.println (".xl142{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;background:yellow;mso-
pattern:auto none;}");
                p.println (".xl143{mso-style-
parent:style0;color:black;font-family:Verdana, sans-
serif;mso-font-charset:0;text-align:left;border-
top:none;border-right:.5pt solid windowtext;border-
bottom:.5pt solid windowtext;border-
left:none;background:lime;mso-pattern:auto none;}.xl144{mso-
style-parent:style0;font-family:Verdana, sans-serif;mso-
font-charset:0;text-align:left;border-top:none;border-
right:.5pt solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;}.xl145{mso-style-
parent:style0;font-family:Verdana, sans-serif;mso-font-
```

```
charset:0;text-align:left;border-top:none;border-right:.5pt
solid windowtext;border-bottom:.5pt solid
windowtext;border-left:none;background:lime;mso-
pattern:auto none;}-->");
                p.println ("</style><!--[if gte mso
9]><xml><x:ExcelWorkbook><x:ExcelWorksheets><x:ExcelWorkshe
et><x:Name>");
                p.print   (goal_label +
"</x:Name><x:WorksheetOptions><x:Print><x:ValidPrinterInfo/
><x:HorizontalResolution>600</x:HorizontalResolution><x:Ver
ticalResolution>600</x:VerticalResolution></x:Print><x:Sele
cted/><x:DoNotDisplayGridlines/><x:Panes><x:Pane><x:Number>
3</x:Number><x:ActiveRow>3</x:ActiveRow></x:Pane></x:Panes>
<x:ProtectContents>False</x:ProtectContents><x:ProtectObjec
ts>False</x:ProtectObjects><x:ProtectScenarios>False</x:Pro
tectScenarios></x:WorksheetOptions></x:ExcelWorksheet></x:E
xcelWorksheets><x:WindowHeight>12270</x:WindowHeight><x:Win
dowWidth>14955</x:WindowWidth><x:WindowTopX>720</x:WindowTo
pX><x:WindowTopY>345</x:WindowTopY><x:ProtectStructure>Fals
e</x:ProtectStructure><x:ProtectWindows>False</x:ProtectWin
dows></x:ExcelWorkbook></xml><![endif]--></head>");

                p.println ("<body link=blue
vlink=purple><table x:str border=0 cellpadding=0
cellspacing=0 width=5801 style='border-collapse:
collapse;table-layout:fixed;width:4335pt'><col width=99
style='mso-width-source:userset;mso-width-
alt:3620;width:74pt'> <col width=75 span=2 style='mso-
width-source:userset;mso-width-alt:2742;width:56pt'> <col
width=138 style='mso-width-source:userset;mso-width-
alt:5046;width:104pt'> <col width=75 style='mso-width-
source:userset;mso-width-alt:2742;width:56pt'> <col
width=104 style='mso-width-source:userset;mso-width-
alt:3803;width:78pt'> <col width=81 style='mso-width-
source:userset;mso-width-alt:2962;width:61pt'> <col
width=94 style='mso-width-source:userset;mso-width-
alt:3437;width:71pt'> <col width=88 style='mso-width-
source:userset;mso-width-alt:3218;width:66pt'><col width=97
style='mso-width-source:userset;mso-width-
alt:3547;width:73pt'> <col width=75 span=65 style='mso-
width-source:userset;mso-width-alt:2742; width:56pt'>");
                p.println (" <tr height=13 style='mso-
height-source:userset;height:9.75pt'>  <td colspan=75
height=13 class=xl24 width=5801 style='height:12.75pt;
width:4335pt'");
                p.print ("x:str=\"Goal: " + goal_statement
+ "\">Goal: ");
```

```
                p.println ("</td></tr><tr height=13
style='mso-height-source:userset;height:9.75pt'><td
height=13 colspan=75 class=xl24 style='height:12.75pt;mso-
ignore:colspan'></td></tr>");
                p.println ("<tr height=13 style='mso-
height-source:userset;height:9.75pt'><td colspan=75
height=13 class=xl24 style='height:12.75pt'>" + SPMC +
"</td><td colspan=72 class=xl24 style='mso-
ignore:colspan'></td></tr>");
                p.print ("<tr height=13 style='mso-height-
source:userset;height:9.75pt'><td height=13 class=xl24
align=right style='height:12.75pt' x:num=\"");
                p.print(PMC + "\">" + PMC);
                p.print ("</td><td colspan=74 class=xl24
style='mso-ignore:colspan'></td></tr><tr height=13
style='mso-height-source:userset;height:12.75pt'><td
height=13 colspan=75 class=xl24 style='height:12.75pt;mso-
ignore:colspan'></td></tr>");
                p.println ("<tr height=13 style='mso-
height-source:userset;height:9.75pt'><td colspan=75
height=13 class=xl24 style='height:12.75pt'>" );
                p.println ("Heading Frequency: " +
job_head_freq + "\tHeading Cosine: " + job_head_cos +
"</td><td colspan=72 class=xl24 style='mso-
ignore:colspan'></td></tr>");
                p.println ("<tr height=13 style='mso-
height-source:userset;height:9.75pt'><td colspan=75
height=13 class=xl24 style='height:12.75pt'>" );
                p.println ("Link Frequency: " +
job_link_freq + "\tLink Cosine: " + job_link_cos +
"</td><td colspan=72 class=xl24 style='mso-
ignore:colspan'></td></tr>");
                p.println ("<tr height=13 style='mso-
height-source:userset;height:9.75pt'><td colspan=75
height=13 class=xl24 style='height:12.75pt'>" );
                p.println ("Space: " + job_space +
"</td><td colspan=72 class=xl24 style='mso-
ignore:colspan'></td></tr>");
                p.println ("<tr height=13 style='mso-
height-source:userset;height:9.75pt'><td colspan=75
height=13 class=xl24 style='height:12.75pt'>" );
                p.println ("Webpage: " + web_label +
"</td><td colspan=72 class=xl24 style='mso-
ignore:colspan'></td></tr>");
                p.println ("<tr height=13 style='mso-
height-source:userset;height:9.75pt'><td height=13
colspan=75 class=xl24 style='height:12.75pt;mso-
```

```
ignore:colspan'></td></tr><tr class=xl25 height=68
style='height:51.0pt'><td height=68 class=xl27 width=120
style='height:51.0pt;width:74pt'>Original Label</td><td
height=68 class=xl27 width=99
style='height:51.0pt;width:74pt'>Text</td>");
                p.println ("<td class=xl28 width=75
style='width:56pt'>Cosine</td><td class=xl28 width=50
style='width:40pt'>Term Vector</td><td class=xl28 width=75
style='width:56pt'>Heading Or Link</td><td class=xl28
width=140 style='width:104pt'>Specific Heading</td><td
class=xl28 width=75 style='width:56pt'>Correct</td><td
class=xl29 width=104 style='width:78pt'>Weak-Scent Correct
Link</td><td class=xl28 width=81
style='width:61pt'>Unfamiliar Correct Link</td><td
class=xl28 width=94 style='width:71pt'>Competing Link under
Competing Heading</td><td class=xl28 width=88
style='width:66pt'>Competing Link under Correct
Heading</td><td class=xl28 width=97
style='width:73pt'>Competing Heading</td><td colspan=65
class=xl26 style='mso-ignore:colspan'></td></tr>");

                int ordered_headings_idx[] = new int
[num_of_headings];
                int ordered_links_idx[] = new int
[num_of_links];

                //sort headings by cosine value
                for (int c=0; c < num_of_headings; c++)
                {
                  if (c==0)
                  {
                    double max_cosine=-1.0;
                    int max_index=0;
                    for (int d=0; d < result_count; d++)
                    {
                      Double cs = new Double (cosine[d]);
                      if (cs.doubleValue() > max_cosine &&
heading_or_link[d].equals("Heading"))
                      {
                        max_cosine=cs.doubleValue();
                        max_index=d;
                      }//end if
                    }//end for
                    ordered_headings_idx[c]=max_index;
                    used[max_index] = "Used";
                  }//end if
                  else
```

```
                    {
                      int local_max_idx = 0;
                      double local_max = -1.0;
                      for (int d=0; d < result_count; d++)
                      {
                        Double cs = new Double (cosine[d]);
                        if
(heading_or_link[d].equals("Heading"))
                        {
                          if (cs.doubleValue() >= local_max
&& !used[d].equals("Used"))
                          {
                            local_max_idx=d;
                            Double lm = new
Double(cosine[local_max_idx]);
                            local_max = lm.doubleValue();
                          }//end if
                        }//end if
                      }//end for
                      ordered_headings_idx[c]=local_max_idx;
                      used[local_max_idx]="Used";
                    }//end else
                  }//end for

                  //sort links
                  for (int c=0; c < num_of_links; c++)
                  {
                    if (c==0)
                    {
                      double max_cosine=-1.0;
                      int max_index=0;
                      for (int d=0; d < result_count; d++)
                      {
                        Double cs = new Double (cosine[d]);
                        if (cs.doubleValue() > max_cosine &&
heading_or_link[d].equals("Link"))
                        {
                          max_cosine=cs.doubleValue();
                          max_index=d;
                        }//end if
                      }//end for
                      ordered_links_idx[c]=max_index;
                      used[max_index] = "Used";
                    }//end if
                    else
                    {
                      int local_max_idx = 0;
```

```
                    double local_max = -1.0;
                    for (int d=0; d < result_count; d++)
                    {
                      Double cs = new Double (cosine[d]);
                      if
(heading_or_link[d].equals("Link"))
                      {
                        if (cs.doubleValue() >= local_max
&& !used[d].equals("Used"))
                        {
                          local_max_idx=d;
                          Double lm = new
Double(cosine[local_max_idx]);
                          local_max = lm.doubleValue();
                        }//end if
                      }//end if
                    }//end for
                    ordered_links_idx[c]=local_max_idx;
                    used[local_max_idx]="Used";
                  }//end else
                }//end for


                //output headings
               for (int c=0; c < num_of_headings; c++)
                {
                  int idx = ordered_headings_idx[c];
                  String sub=null;
                  String whole = null;

                  if (elaborated_label[idx].length()>14)
                  {
                    sub =
elaborated_label[idx].substring(0,14);
                    whole =
elaborated_label[idx].substring(14);
                  }
                  else
                  {
                    sub =
elaborated_label[idx].substring(0,elaborated_label[idx].len
gth());
                    whole =
elaborated_label[idx].substring(elaborated_label[idx].lengt
h());
                  }
```

```
                    if (correct[idx].equals("Yes"))
                    {
                      p.println("<tr height=17
style='height:12.75pt'>");
                      p.println("<td class=xl41>" +
original_label[idx] + "</td>");
                      p.print ("<td height=17 class=xl41
style='height:12.75pt'>");
                      p.print( sub + "<span
style='display:none'>");
                      p.println(whole + "</span></td>");
                      p.println("<td class=xl41 x:num>" +
cosine[idx] + "</td>");
                      if (term_vectors[idx]==99999)
                        p.println("<td class=xl41 x:num>Can't
find any terms from text</td>");
                      else
                        p.println("<td class=xl41 x:num>" +
term_vectors[idx] + "</td>");
                      p.println("<td
class=xl41>Heading</td>");
                      p.println("<td class=xl41>" + sub +
"</td>");
                      p.println("<td class=xl41>X</td>");

                      if
(classification[idx][0].equals(" "))
                        p.println("<td
class=xl41> </td>");
                      else
                        p.println("<td class=xl41>X</td>");

                      if
(classification[idx][1].equals(" "))
                        p.println("<td
class=xl41> </td>");
                      else
                        p.println("<td class=xl41>X</td>");

                      if
(classification[idx][2].equals(" "))
                        p.println("<td
class=xl41> </td>");
                      else
                        p.println("<td class=xl41>X</td>");
```

```
                    if
(classification[idx][3].equals(" "))
                      p.println("<td
class=xl141> </td>");
                    else
                      p.println("<td class=xl141>X</td>");

                    if
(classification[idx][4].equals(" "))
                      p.println("<td
class=xl141> </td>");
                    else
                      p.println("<td class=xl141>X</td>");
                  }
                  else if
(classification[idx][0].equals(" ") &&
classification[idx][1].equals(" ") &&
classification[idx][2].equals(" ") &&
classification[idx][3].equals(" ") &&
classification[idx][4].equals(" "))
                  {
                    p.println("<tr height=17
style='height:12.75pt'>");
                    p.println("<td class=xl140>" +
original_label[idx] + "</td>");
                    p.println("<td height=17 class=xl140
style='height:12.75pt'>");
                    p.print( sub + "<span
style='display:none'>");
                    p.println(whole + "</span></td>");
                    p.println("<td class=xl40 x:num>" +
cosine[idx] + "</td>");
                    if (term_vectors[idx]==99999)
                      p.println("<td class=xl40 x:num>Can't
find any terms from text</td>");
                    else
                      p.println("<td class=xl40 x:num>" +
term_vectors[idx] + "</td>");
                    p.println("<td
class=xl40>Heading</td>");
                    p.println("<td class=xl40>" + sub +
"</td>");
                    p.println("<td
class=xl40> </td>");
                    p.println("<td
class=xl40> </td>");
```

```
                    p.println("<td
class=xl140> </td>");
                    p.println("<td
class=xl140> </td>");
                    p.println("<td
class=xl140> </td>");
                    p.println("<td
class=xl140> </td>");
                  }
                  else
                  {
                    p.println("<tr height=17
style='height:12.75pt'>");
                    p.println("<td class=xl130>" +
original_label[idx] + "</td>");
                    p.println("<td height=17 class=xl138
style='height:12.75pt'>");
                    p.print( sub + "<span
style='display:none'>");
                    p.println(whole + "</span></td>");
                    p.println("<td class=xl130 x:num>" +
cosine[idx] + "</td>");
                    if (term_vectors[idx]==99999)
                      p.println("<td class=xl130 x:num>Can't
find any terms from text</td>");
                    else
                      p.println("<td class=xl130 x:num>" +
term_vectors[idx] + "</td>");
                    p.println("<td
class=xl130>Heading</td>");
                    p.println("<td class=xl142>" + sub +
"</td>");

                    p.println("<td
class=xl130> </td>");
                    if
(classification[idx][0].equals(" "))
                      p.println("<td
class=xl131> </td>");
                    else
                      p.println("<td class=xl130>X</td>");

                    if
(classification[idx][1].equals(" "))
                      p.println("<td
class=xl130> </td>");
                    else
```

```
                    p.println("<td class=xl30>X</td>");

                  if
(classification[idx][2].equals(" "))
                    p.println("<td
class=xl30> </td>");
                  else
                    p.println("<td class=xl30>X</td>");

                  if
(classification[idx][3].equals(" "))
                    p.println("<td
class=xl30> </td>");
                  else
                    p.println("<td class=xl30>X</td>");

                  if
(classification[idx][4].equals(" "))
                    p.println("<td
class=xl30> </td>");
                  else
                    p.println("<td class=xl30>X</td>");
                }//end else if
                p.println ("<td colspan=65 class=xl24
style='mso-ignore:colspan'></td></tr>");
              }

              // Output Link Data
              for (int c=0; c < num_of_links; c++)
              {
                int idx = ordered_links_idx[c];
                String sub=null;
                String whole = null;
                String subl=null;
                if (elaborated_label[idx].length()>14)
                {
                  sub =
elaborated_label[idx].substring(0,14);
                  whole =
elaborated_label[idx].substring(14);
                }
                else
                {
                  sub =
elaborated_label[idx].substring(0,
elaborated_label[idx].length());
```

```
                    whole =
elaborated_label[idx].substring(elaborated_label[idx].lengt
h());
                  }

                  if (specific_heading[idx].length()>14)
                    subl =
specific_heading[idx].substring(0,14);
                  else
                    subl =
specific_heading[idx].substring(0,
specific_heading[idx].length());

                  if (correct[idx].equals("Yes"))
                  {
                    p.println("<tr height=17
style='height:12.75pt'>");
                    p.println("<td class=xl141>" +
original_label[idx] + "</td>");
                    p.println("<td height=17 class=xl141
style='height:12.75pt'>");
                    p.print( sub + "<span
style='display:none'>");
                    p.println(whole + "</span></td>");
                    p.println("<td class=xl41 x:num>" +
cosine[idx] + "</td>");
                    if (term_vectors[idx]==99999)
                      p.println("<td class=xl41 x:num>Can't
find any terms from text</td>");
                    else
                      p.println("<td class=xl41 x:num>" +
term_vectors[idx] + "</td>");
                    p.println("<td class=xl41>Link</td>");
                    p.println("<td class=xl41>" + subl +
"</td>");
                    p.println("<td class=xl41>X</td>");
                    if
(classification[idx][0].equals(" "))
                      p.println("<td
class=xl41> </td>");
                    else
                      p.println("<td class=xl41>X</td>");

                    if
(classification[idx][1].equals(" "))
                      p.println("<td
class=xl41> </td>");
```

```
                    else
                      p.println("<td class=xl41>X</td>");

                    if
(classification[idx][2].equals(" "))
                      p.println("<td
class=xl41> </td>");
                    else
                      p.println("<td class=xl41>X</td>");

                    if
(classification[idx][3].equals(" "))
                      p.println("<td
class=xl41> </td>");
                    else
                      p.println("<td class=xl41>X</td>");

                    if
(classification[idx][4].equals(" "))
                      p.println("<td
class=xl41> </td>");
                    else
                      p.println("<td class=xl41>X</td>");
                  }
                  else if
(classification[idx][0].equals(" ") &&
classification[idx][1].equals(" ") &&
classification[idx][2].equals(" ") &&
classification[idx][3].equals(" ") &&
classification[idx][4].equals(" "))
                  {
                    p.println("<tr height=17
style='height:12.75pt'>");
                    p.println("<td class=xl40>" +
original_label[idx] + "</td>");
                    p.println("<td height=17 class=xl40
style='height:12.75pt'>");
                    p.print( sub + "<span
style='display:none'>");
                    p.println(whole + "</span></td>");
                    p.println("<td class=xl40 x:num>" +
cosine[idx] + "</td>");
                    if (term_vectors[idx]==99999)
                      p.println("<td class=xl40 x:num>Can't
find any terms from text</td>");
                    else
```

```
                        p.println("<td class=xl140 x:num>" +
term_vectors[idx] + "</td>");
                        p.println("<td class=xl140>Link</td>");
                        p.println("<td class=xl140>" + subl +
"</td>");
                        p.println("<td
class=xl140> </td>");
                        p.println("<td
class=xl140> </td>");
                        p.println("<td
class=xl140> </td>");
                        p.println("<td
class=xl140> </td>");
                        p.println("<td
class=xl140> </td>");
                        p.println("<td
class=xl140> </td>");
                      }
                      else
                      {
                        p.println("<tr height=17
style='height:12.75pt'>");
                        p.println("<td class=xl130>" +
original_label[idx] + "</td>");
                        p.println("<td height=17 class=xl130
style='height:12.75pt'>");

                        p.print( sub + "<span
style='display:none'>");
                        p.println(whole + "</span></td>");
                        p.println("<td class=xl130 x:num>" +
cosine[idx] + "</td>");
                        if (term_vectors[idx]==99999)
                          p.println("<td class=xl130 x:num>Can't
find any terms from text</td>");
                        else
                          p.println("<td class=xl130 x:num>" +
term_vectors[idx] + "</td>");
                        p.println("<td class=xl130>Link</td>");
                        p.println("<td class=xl142>" + subl +
"</td>");

                        p.println("<td
class=xl130> </td>");
                        if
(classification[idx][0].equals(" "))
```

```
                        p.println("<td
class=xl31> </td>");
                      else
                        p.println("<td class=xl30>X</td>");

                      if
(classification[idx][1].equals(" "))
                        p.println("<td
class=xl30> </td>");
                      else
                        p.println("<td class=xl30>X</td>");

                      if
(classification[idx][2].equals(" "))
                        p.println("<td
class=xl30> </td>");
                      else
                        p.println("<td class=xl30>X</td>");

                      if
(classification[idx][3].equals(" "))
                        p.println("<td
class=xl30> </td>");
                      else
                        p.println("<td class=xl30>X</td>");

                      if
(classification[idx][4].equals(" "))
                        p.println("<td
class=xl30> </td>");
                      else
                        p.println("<td class=xl30>X</td>");
                    }//end else if
                    p.println ("<td colspan=65 class=xl24
style='mso-ignore:colspan'></td></tr>");
                  }

                  p.println ("<![if
supportMisalignedColumns]> <tr height=0
style='display:none'>  <td width=99
style='width:74pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=138
style='width:104pt'></td>  <td width=75
style='width:56pt'></td>  <td width=104
style='width:78pt'></td>  <td width=81
style='width:61pt'></td>  <td width=94
```

```
style='width:71pt'></td>  <td width=88
style='width:66pt'></td>  <td width=97
style='width:73pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
```

```
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>");
                p.println ("<td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td>  <td width=75
style='width:56pt'></td> </tr>
<![endif]></table></body></html>");

                p.close();
              }
              catch (Exception e){e.printStackTrace();}
            }//end for int b

            try
            {
              String command = "/usr2/home/brownr/email-
rep.sh /tmp/acww_job" + args[a] + " " + job_email + "
\"ACWW Results\" ACWW";
               Process ls_proc =
Runtime.getRuntime().exec(command);
              try
              {
                Thread.sleep(100000);//wait_time);
              }
```

```
                catch(InterruptedException
e){System.out.println("Sleep Interrupted");}
                command = "rm -rf /tmp/acww_job" + args[a] +
"/*;rmdir /tmp/acww_job" + args[a];
                 ls_proc =
Runtime.getRuntime().exec(command);
             } catch (IOException e1)
{System.err.println(e1); System.exit(1);}

          }//end for int a
        } catch ( java.sql.SQLException e )
{e.printStackTrace();}
        rs.close();
        stat.close();
        conn.close();

        return 1;
      }//end try
      catch ( java.sql.SQLException e
){e.printStackTrace();}
      catch ( java.lang.ClassNotFoundException e
){e.printStackTrace();}
      catch ( java.lang.IllegalAccessException e
){e.printStackTrace();}
      catch ( java.lang.InstantiationException e
){e.printStackTrace();}

      return 0;
  }

//-------------------------------------------------------------
-----------------------------------------
//           COUNT NUMBER OF WORDS IN A STRING
//-------------------------------------------------------------
-----------------------------------------
     private static long countWords(String line) {
          long numWords = 0;
          int index = 0;
          boolean prevWhitespace = true;
          while (index < line.length()) {
               char c = line.charAt(index++);
               boolean currWhitespace =
Character.isWhitespace(c);
               if (prevWhitespace && !currWhitespace) {
                    numWords++;
               }
               prevWhitespace = currWhitespace;
```

```
            }
            return numWords;
        }


//-------------------------------------------------------------
-----------------------------------------
//           PERFORM ONE2MANY ANALYSIS
//-------------------------------------------------------------
-----------------------------------------
  private int perform_one_to_many(String[] args)
  {
    System.out.println("Performing one2many");
     try
     {

Class.forName("com.mysql.jdbc.Driver").newInstance();
       Connection conn = DriverManager.getConnection(url);
       ResultSet rs = null;
       Statement stat = null;
       String sql = null;
       try
       {
          for (int a=0; a < args.length; a++)
          { //Cycle through the job_num(s) provided by the
php script
            stat = conn.createStatement();
            sql = "SELECT * FROM job_options WHERE
job_num=" + args[a];
            rs = stat.executeQuery(sql);
            rs.next();
            String job_space =  rs.getString("job_space");

            //Grab the webpage and goal to analyze together
            sql = "SELECT * FROM matrix WHERE job_num=" +
args[a];
            rs = stat.executeQuery(sql);
            Vector matrix_num_vector = new Vector();
            Vector matrix_web_num_vector = new Vector();
            Vector matrix_goal_num_vector = new Vector();
            while(rs.next())
            {

matrix_num_vector.add(rs.getString("matrix_num"));

matrix_web_num_vector.add(rs.getString("web_num"));
```

```
matrix_goal_num_vector.add(rs.getString("goal_num"));
            }
            int matrix_count = matrix_num_vector.size();
            String matrix_num[] = new String[matrix_count];
            String matrix_web_num[] = new
String[matrix_count];
            String matrix_goal_num[] = new
String[matrix_count];
            matrix_num_vector.copyInto(matrix_num);
            matrix_web_num_vector.copyInto(matrix_web_num);

matrix_goal_num_vector.copyInto(matrix_goal_num);


            for (int b=0; b < matrix_count; b++)
            {
              sql = "SELECT * FROM goal WHERE goal_num=" +
matrix_goal_num[b];
              rs = stat.executeQuery(sql);
              rs.next();
              String goal_statement =
rs.getString("goal_statement");
              sql = "SELECT * FROM heading WHERE web_num="
+ matrix_web_num[b];
              rs = stat.executeQuery(sql);
              Vector heading_num_vector = new Vector();
              Vector heading_label_vector = new Vector();
              Vector heading_elaborated_label_vector = new
Vector();
              while(rs.next())
              {

heading_num_vector.add(rs.getString("heading_num"));

heading_label_vector.add(rs.getString("heading_label"));

heading_elaborated_label_vector.add(rs.getString("elaborate
d_label"));
              }
              int heading_count =
heading_num_vector.size();
              String heading_num [] = new
String[heading_count];
              String heading_label[] = new
String[heading_count];
```

```
                String heading_elaborated_label[] = new
String [heading_count];
                heading_num_vector.copyInto(heading_num);
                heading_label_vector.copyInto(heading_label);

heading_elaborated_label_vector.copyInto(heading_elaborated
_label);

                Vector correct_heading_num_vector = new
Vector();
                Vector correct_head_num_vector = new
Vector();
                sql = "SELECT * FROM correct_heading WHERE
matrix_num=" + matrix_num[b];
                rs = stat.executeQuery(sql);
                while(rs.next())
                {

correct_heading_num_vector.add(rs.getString("correct_headi_
num"));//correct heading index

correct_head_num_vector.add(rs.getString("head_num"));
//heading index
                }
                int correct_heading_count =
correct_heading_num_vector.size();
                String correct_heading_num[] = new
String[correct_heading_count];
                String correct_head_num[] = new
String[correct_heading_count];

correct_heading_num_vector.copyInto(correct_heading_num);

correct_head_num_vector.copyInto(correct_head_num);

                String heading_cosine = null;
                String ls_str = null;
                //Start working on links
                for (int c=0; c < heading_count; c++)
                {
                  //try
                  int correct_head_index=-1;
                  {
                    boolean correct=false;
                       try
                    {
                      //This calls the elaboration script
```

```
                    String eGoal =
URLEncoder.encode(goal_statement);
                    //eGoal=this.fix(eGoal);
                    String eHL =
URLEncoder.encode(heading_elaborated_label[c]);
                    //eHL=this.fix(eHL);
                    String command =
"/usr2/home/brownr/acww-one2many.cgi Space=" + job_space +
" Goal=" + eGoal + " Links=" + eHL;
                                        Process ls_proc =
Runtime.getRuntime().exec(command);
                      DataInputStream ls_in = new
DataInputStream(ls_proc.getInputStream());
                                        while ((ls_str =
ls_in.readLine()) != null)
                    {
                      StringBuffer sb = new StringBuffer();
                      heading_cosine = ls_str;
System.out.println("Heading Cosine: " + heading_cosine + "
for " + eHL);
                        }
                    for (int z=0; z <
correct_heading_count; z++)
                    {
                      if
(correct_head_num[z].equals(heading_num[c]))
                      {
                        correct=true;
                        correct_head_index=0;
                      }
                    }
                    if(correct==false)
                      sql = "INSERT INTO
one_to_many_results (matrix_num, original_label,
elaborated_label, cosine, heading_or_link,
specific_heading, correct, heading_num, job_num) VALUES
(\""
                         + matrix_num[b] + "\", \"" +
heading_label[c] + "\",\"" + heading_elaborated_label[c] +
"\", \"" + heading_cosine + "\", \"" + "Heading" + "\", \""
+ heading_label[c] + "\", \"No\",\""  + heading_num[c] +
"\",\"" + args[a] + "\")";
                    else
                      sql = "INSERT INTO
one_to_many_results (matrix_num, original_label,
elaborated_label, cosine, heading_or_link,
```

```
specific_heading, correct, heading_num, job_num) VALUES
(\""
                          + matrix_num[b] + "\", \"" +
heading_label[c] + "\",\"" + heading_elaborated_label[c] +
"\", \"" + heading_cosine + "\", \"" + "Heading" + "\", \""
+ heading_label[c] + "\", \"Yes\",\"" + heading_num[c] +
"\",\"" + args[a] + "\")";
                    PreparedStatement ps =
conn.prepareStatement(sql);
                    ps.execute();
                  } catch (IOException e) {System.exit(0);}
                } //catch (IOException e1)
{System.err.println(e1); System.exit(1);}

                sql = "SELECT * FROM link WHERE head_num="
+ heading_num[c];
                rs = stat.executeQuery(sql);
                Vector link_num_vector = new Vector();
                Vector link_label_vector = new Vector();
                Vector link_elaborated_label_vector = new
Vector();
                while(rs.next())
                {

link_num_vector.add(rs.getString("link_num"));

link_label_vector.add(rs.getString("link_label"));

link_elaborated_label_vector.add(rs.getString("elaborated_l
abel"));
                }
                int link_count = link_num_vector.size();
                String link_num [] = new
String[link_count];
                String link_label[] = new
String[link_count];
                String link_elaborated_label[] = new String
[link_count];
                link_num_vector.copyInto(link_num);
                link_label_vector.copyInto(link_label);

link_elaborated_label_vector.copyInto(link_elaborated_label
);

                Vector correct_link_num_vector = null;
                int correct_link_count = 0;
                String correct_link_num[] = null;
```

```
                if (correct_head_index!=-1)
                {
                  correct_link_num_vector = new Vector();
                  sql = "SELECT * FROM correct_link WHERE
correct_head_num=" +
correct_heading_num[correct_head_index];
                  rs = stat.executeQuery(sql);
                  while(rs.next())
                  {

correct_link_num_vector.add(rs.getString("link_num"));
//linking index
                  }
                  correct_link_count =
correct_link_num_vector.size();
                  correct_link_num = new
String[correct_link_count];

correct_link_num_vector.copyInto(correct_link_num);
                }
                for (int d=0; d < link_count; d++)
                {
                   boolean correct=false;
                   String cosine = null;
                   try
                   {
                    ls_str=null;
                    //This calls the elaboration script
                    String eGoal =
URLEncoder.encode(goal_statement);
                    //eGoal=this.fix(eGoal);
                    String eL =
URLEncoder.encode(link_elaborated_label[d]);
                    //eL=this.fix(eL);
                    String command =
"/usr2/home/brownr/acww-one2many.cgi Space=" + job_space +
" Goal=" + eGoal + " Links=" + eL;
                     Process ls_proc =
Runtime.getRuntime().exec(command);
                        DataInputStream ls_in = new
DataInputStream(ls_proc.getInputStream());
                        try
                    {
                         while ((ls_str = ls_in.readLine())
!= null)
                      {
```

```
                        StringBuffer sb = new
StringBuffer();
                        cosine = ls_str;
System.out.println("Cosine: " + cosine + " for " + eL);
                            }

                    } catch (IOException e)
{System.exit(0);}
                    } catch (IOException e1)
{System.err.println(e1); System.exit(1);}

                    if (correct_head_index!=-1)
                    {
                      for (int z=0; z < correct_link_count;
z++)
                      {
                        if
(correct_link_num[z].equals(link_num[d]))
                            correct=true;
                      }
                    }
                    if(correct==false)
                       sql = "INSERT INTO
one_to_many_results (matrix_num, original_label,
elaborated_label, cosine, heading_or_link,
specific_heading, correct, heading_num, job_num) VALUES
(\""
                            + matrix_num[b] + "\", \"" +
link_label[d] + "\",\"" + link_elaborated_label[d] + "\",
\"" + cosine + "\", \"" + "Link" + "\", \"" +
heading_label[c] + "\", \"No\",\"" + heading_num[c] +
"\",\"" + args[a] + "\")";
                    else
                       sql = "INSERT INTO
one_to_many_results (matrix_num, original_label,
elaborated_label, cosine, heading_or_link,
specific_heading, correct, heading_num, job_num) VALUES
(\""
                            + matrix_num[b] + "\", \"" +
link_label[d] + "\",\"" + link_elaborated_label[d] + "\",
\"" + cosine + "\", \"" + "Link" + "\", \"" +
heading_label[c] + "\", \"Yes\",\"" + heading_num[c] +
"\",\"" + args[a] + "\")";
                     PreparedStatement ps =
conn.prepareStatement(sql);
                     ps.execute();
                  }//end for int d
```

```
              }//end for int c
            }//end for int b
          }//end for int a
        } catch ( java.sql.SQLException e )
{e.printStackTrace();}
        rs.close();
        stat.close();
        conn.close();
        return 1;
      }//end try
      catch ( java.sql.SQLException e
){e.printStackTrace();}
      catch ( java.lang.ClassNotFoundException e
){e.printStackTrace();}
      catch ( java.lang.IllegalAccessException e
){e.printStackTrace();}
      catch ( java.lang.InstantiationException e
){e.printStackTrace();}

      return 0;
  }


//-------------------------------------------------------
-----------------------------------------
//           PERFORM HEADING/LINK ELABORATION
//-------------------------------------------------------
-----------------------------------------
  private int elaborate_headings_and_links(String[] args)
  {
    System.out.println("Performing elaboration");
     try
     {

Class.forName("com.mysql.jdbc.Driver").newInstance();
        Connection conn = DriverManager.getConnection(url);
        ResultSet rs = null;
        Statement stat = null;
        String sql = null;
        try
        {
           for (int a=0; a < args.length; a++)
           { //Cycle through the job_num(s) provided by the
php script
             stat = conn.createStatement();
             sql = "SELECT * FROM job_options WHERE
job_num=" + args[a];
```

```
            rs = stat.executeQuery(sql);
            rs.next();
            String job_space =  rs.getString("job_space");
            float job_link_freq =
rs.getFloat("job_link_freq");
            float job_link_cos =
rs.getFloat("job_link_cosine");
            float job_head_freq =
rs.getFloat("job_head_freq");
            float job_head_cos =
rs.getFloat("job_head_cosine");
             //Pull Webpage Information from SQL database
            Vector web_vector = new Vector();
            sql = "SELECT * FROM webpage WHERE job_num = "
+ args[a];
            rs = stat.executeQuery(sql);
            while(rs.next())
            {
              web_vector.add(rs.getString("web_num"));
            }
            int web_count = web_vector.size();
            String web[] = new String[web_count];
            web_vector.copyInto(web);

            for (int z=0; z < web_count; z++)
            {    //Pull heading information from SQL
database
              sql = "SELECT * FROM heading WHERE web_num =
" + web[z];
              rs = stat.executeQuery(sql);
              Vector heading_vector = new Vector();
              Vector heading_label_vector = new Vector();
              while(rs.next())
              {

heading_vector.add(rs.getString("heading_num"));

heading_label_vector.add(rs.getString("heading_label"));
              }
              int heading_count = heading_vector.size();
              String headings[] = new
String[heading_count];
              String heading_label[] = new
String[heading_count];
              heading_label_vector.copyInto(heading_label);
              heading_vector.copyInto(headings);
              for (int x=0; x< heading_count; x++)
```

```
                {
                  //Pull link information from SQL database
                  sql = "SELECT * FROM link WHERE head_num="
+ headings[x];
                  rs = stat.executeQuery(sql);
                  Vector link_vector = new Vector();
                  Vector link_num_vector = new Vector();
                  while (rs.next())
                  {

link_vector.add(rs.getString("link_label"));

link_num_vector.add(rs.getString("link_num"));
                  }
                  int link_count = link_vector.size();
                  String links[] = new String[link_count];
                  String link_num[] = new String[link_count];
                  link_vector.copyInto(links);
                  link_num_vector.copyInto(link_num);

                  String heading_elaboration = "";
                  StringBuffer sbhe = new StringBuffer();
                  sbhe.equals(heading_elaboration);

                  try
                  {
                    if (job_head_freq==-2.0 ||
job_head_freq==-2)
                    { //No Heading Elaboration
                      sbhe.append(heading_label[x]);
                    }
                    else if (job_head_freq == -1.0 ||
job_head_freq == -1)
                    { //Minimum Heading Elaboration
                      sbhe.append(heading_label[x]);
                      for (int y=0; y < link_count; y++)
                      {
                        sbhe.append(" ");
                        sbhe.append(links[y]);
                      }
                    }
                    else if (job_head_freq > -1)
                    { //Heading Elaboration with specified
frequency & cosine
                      String ls_str;
                      //This calls the elaboration script
```

```
                    String eHL =
URLEncoder.encode(heading_label[x]);
                    //eHL=this.fix(eHL);
                    String command =
"/usr2/home/brownr/acww-elaborate.cgi Space=" + job_space +
" Frequency=" + job_head_freq + " Cosine=" + job_head_cos +
" Links=" + eHL;
                     Process ls_proc =
Runtime.getRuntime().exec(command);
                        DataInputStream ls_in = new
DataInputStream(ls_proc.getInputStream());
                        try
                    {
                         while ((ls_str = ls_in.readLine())
!= null)
                       {
                         sbhe.append(" ");
                         sbhe.append(ls_str);
                             }
                    } catch (IOException e)
{System.exit(0);}
System.out.println("Command: " + command);
System.out.println("Elaboration: " + sbhe.toString());
                 }//end else
                } catch (IOException e1)
{System.err.println(e1); System.exit(1);}

               for (int y=0; y< link_count; y++)
               {   //Cycle through links to perform link
elaboration
                 String elaboration = "";
                 if (job_link_freq == -2.0 ||
job_link_freq == -2)
                 {
                     elaboration = links[y];
                 }
                 else
                 {
                   try
                   {
                     String ls_str;
                     //Run elaboration.  acww-elaborate
Space=  Frequency=50, Cosine=.50 Links=Music
                     String eHL =
URLEncoder.encode(links[y]);
                     //eHL=this.fix(eHL);
```

```
                        String command
="/usr2/home/brownr/acww-elaborate.cgi Space=" + job_space
+ " Frequency=" + job_link_freq + " Cosine=" + job_link_cos
+ " Links=" + eHL;
                          // get its output (your input)
stream
                           Process ls_proc =
Runtime.getRuntime().exec(command);
                        DataInputStream ls_in = new
DataInputStream(ls_proc.getInputStream());
                        try
                         {
                           StringBuffer sb = new
StringBuffer();
//                           sb.append(elaboration);
                               while ((ls_str =
ls_in.readLine()) != null)
                           {
//                             sb.append(" ");
                           sb.append(ls_str);
                           }
                           for (int ch=0; ch < sb.length();
ch++)
                           {

                             if
(Character.isWhitespace(sb.charAt(ch)) && (ch==0 ||
ch==sb.length()) )
                               sb.deleteCharAt(ch);
                             if
(Character.isUpperCase(sb.charAt(ch)) )
                               sb.setCharAt(ch,
Character.toLowerCase(sb.charAt(ch)));
                             if
(!Character.isLetter(sb.charAt(ch)) &&
!Character.isWhitespace(sb.charAt(ch)))
                               sb.deleteCharAt(ch);
                             if ( ch >0)
                             {
                               if
(Character.isWhitespace(sb.charAt(ch-1)) &&
Character.isWhitespace(sb.charAt(ch)))
                                 {
                                   sb.deleteCharAt(ch);
                                 }
                             }
                           }
```

```
                        elaboration = sb.toString();
System.out.println("Command: " + command);
System.out.println("Elaboration: " + elaboration);
                    } catch (IOException e)
{System.exit(0);}
                     } catch (IOException e1)
{System.err.println(e1); System.exit(1);}
                  }//end else

                  if (job_head_freq > -1 || job_head_freq >
-1.0)
                     sbhe.append(" " + elaboration);
                  //Save elaboration
                  sql = "UPDATE link SET
elaborated_label=\"" + elaboration + "\" WHERE link_num=" +
link_num[y];
                  PreparedStatement ps =
conn.prepareStatement(sql);
                  ps.execute();
                }//end for int y

                for (int ch=0; ch < sbhe.length(); ch++)
                {
                  if
(Character.isWhitespace(sbhe.charAt(ch)) && (ch==0 ||
ch==sbhe.length()) )
                    sbhe.deleteCharAt(ch);
                  if
(Character.isUpperCase(sbhe.charAt(ch)) )
                    sbhe.setCharAt(ch,
Character.toLowerCase(sbhe.charAt(ch)));
                  if (!Character.isLetter(sbhe.charAt(ch))
&& !Character.isWhitespace(sbhe.charAt(ch)))
                    sbhe.deleteCharAt(ch);
                  if ( ch >0)
                  {
                    if
(Character.isWhitespace(sbhe.charAt(ch-1)) &&
Character.isWhitespace(sbhe.charAt(ch)))
                      sbhe.deleteCharAt(ch);
                  }
                }
              heading_elaboration = sbhe.toString();
              sql = "UPDATE heading SET
elaborated_label=\"" + heading_elaboration + "\" WHERE
heading_num=" + headings[x];
```

```
                PreparedStatement ps =
conn.prepareStatement(sql);
                ps.execute();

              }//end for int x=0
            }//end for int z
          }//end for int a
        } catch ( java.sql.SQLException e )
{e.printStackTrace();}
        rs.close();
        stat.close();
        conn.close();

        return 1;
      }//end try
      catch ( java.sql.SQLException e
){e.printStackTrace();}
      catch ( java.lang.ClassNotFoundException e
){e.printStackTrace();}
      catch ( java.lang.IllegalAccessException e
){e.printStackTrace();}
      catch ( java.lang.InstantiationException e
){e.printStackTrace();}

      return 0;
  }//end private elaborate_headings_and_links

  //Main method
  public static void main(String[] args)
  {
    new backend(args);
  }

}
```

C.4. AutoCWW.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<title>ACWW Home Page</title>
</head>

<body bgcolor="#99CCFF" text="#000033">
<table width="950" border="1" cellspacing="8"
cellpadding="8" bgcolor="#3333FF">
  <tr>
    <td><h4><font color="#99CCFF" size="7" face="Arial,
Helvetica, sans-serif">ACWW</font><font color="#99CCFF"
size="+7" face="Arial, Helvetica, sans-serif"><br>
        </font><font color="#99CCFF" size="3" face="Arial,
Helvetica, sans-serif">A
        joint project of Richard Brown, graduate student,
Department of Computer
        and Information Sciences, University of North
Florida and AutoCWW research
        program, Institute of Cognitive Science (ICS),
University of Colorado
        at Boulder<br>
        </font></h4>
      <h4><font color="#99CCFF" size="2" face="Arial,
Helvetica, sans-serif"><em>ACWW
        webmaster: Richard Brown<br>
        University of North Florida. Jacksonville, FL<br>
        lxadu99b@comcast.net, phone 904-220-
5224</em></font><em></p> </em> </h4>
      <h4><font color="#99CCFF" size="2" face="Arial,
Helvetica, sans-serif">AutoCWW
            webmaster: Marilyn Hughes Blackmon, Ph.D. <br>
        Research Associate, Institute of Cognitive Science.
University
of Colorado, Boulder,
  CO 80309-0344<br>
        blackmon@psych.colorado.edu, phone 303-859-5060,
fax 303-492-7177<br>
        </font><em><font color="#99CCFF" size="2"
face="Arial, Helvetica, sans-serif">AutoCWW
        Research Program: This material is based upon work
supported by the National
        Science Foundation under Grant No. 0137759. Any
opinions, findings, and
        conclusions or recommendations expressed in this
material are those of
```

```
        the author(s) and do not necessarily reflect the
views of the National
        Science Foundation.</font></em></h4>
    </td>
  </tr>
</table>
<p><font size="4" face="Times New Roman, Times, serif">This
is the home
  page for Richard Brown's ACWW website. For his M.S.
Thesis, Richard has built a new interface called ACWW
  that fully automates the multistep CWW process for an
individual webpage. This
  additional automation of CWW will benefit HCI researchers
and web developers
  who wish to use CWW but find it cumbersome to do CWW on
the current web-based
  interface at http://AutoCWW.colorado.edu. </font></p>
<p> </p>
<p><font size="5" face="Times New Roman, Times, serif">The
current version of ACWW can be located at <a
href="ACWW.php">http://autocww.colorado.edu/~brownr/ACWW.ph
p</a>. </font></p>
<p><font color="#FF0000" size="4" face="Times New Roman,
Times, serif"><strong>Warning:
  As of January 14, 2005, ACWW is still undergoing rigorous
checks for accuracy
  in the output files emailed to users compared to the more
time-consuming CWW
  that users can perform at <a
href="http://autocww.colorado.edu">http://autocww.colorado.
edu</a>.
  The ACWW interface is also getting modifications to
improve usability. </strong></font><font color="#FF0000"
size="4" face="Times New Roman, Times,
serif"><strong>Please
  monitor this webpage periodically for updates on the ACWW
interface, the tutorial,
  the evaluation of ACWW accuracy, and a link to download
Richard Brown's M.S.
  thesis when it is completed soon. </strong></font></p>
<p> </p>
<p><font size="5" face="Times New Roman, Times, serif"><a
href="ACWWTutorial0501123.doc">Download
  latest version of the ACWW Tutorial</a>. </font></p>
<p> </p>
</body>
</html>
```

C.5. Finished.html

```
<html>
<head>
<title>ACWW: Step5</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong><font color="#FFFFFF">Step
                      2: Enter the Webpage's Links &amp;
Headings</font></strong></font><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div align="center"><font
color="#FFFFFF"><strong>Step
                      3: Group Goals and
WebPages</strong></font></div></td>
                  <td width="135"><div
align="center"><strong><font color="#FFFFFF">Step
                      4: Identify Heading &amp;
Links</font></strong></div></td>
                  <td width="135"><strong><font
color="#FF0000">Step 5:Finish</font></strong></td>
                </tr>
              </table>
            </div></td>
```

```
          <td width="1%"></td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
        <tr>
          <td width="1%" bgcolor="#0000CC"> </td>
          <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="">
            <div align="center">
              <p><font size="+3">Thank you for using ACWW
for your CWW needs.
                Your CWW request has been submited into
the queue. As soon as
                it has been processed, the results will
be emailed to the
                address that you provided.</font></p>
              <p> </p>
              <p><font size="+3">Click <a
href="ACWW.php">here</a> to run ACWW again.</font></p>
            </div>
            <blockquote>
              <p> </p>
              </blockquote>
          </form></td>
          <td width="1%" bgcolor="#0000CC"> </td>
        </tr>
      </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
</body>
</html>
```

C.6. Step1.html

```
<html>
<head>
<title>ACWW: Step1</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FF0000">Step
                        1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"></font></div></td>
                  <td width="135"> </td>
                  <td width="135"> </td>
                </tr>
              </table>
            </div></td>
          <td width="1%"></td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
        <tr>
          <td width="1%" bgcolor="#0000CC"> </td>
```

```
          <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="ACWW.php">
              <blockquote>
                <p><br>
                </p>
                <p><b>1. Enter a Label (a unique
combination of number or letters)
                  for the Goal Statement:</b><br>
                  <input name="label" type="text"
id="label" value=""  size="80" maxlength="150">
                </p>
                    <p><b>2. Enter a unique filename to
output the results:<br>
                  </b>
                  <input name="filename" type="text"
id="filename" value=""  size="80" maxlength="150">
                </p>
                <p><strong>3. Enter a Goal Statement:<br>
                  <textarea name="goalstatement" cols="100"
rows="15" id="goalstatement"></textarea>
                  </strong></p>
                <p> </p>
                <p align="center">
                  <input name="another" type="submit"
id="another" value="Enter Another Goal Statment">
                  <input name="next" type="submit"
id="next" value="Next Step">
                </p>
              </blockquote>
            </form></td>
          <td width="1%" bgcolor="#0000CC"> </td>
        </tr>
      </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
<p> </p>
<h2 align="center"><font color="#FFFFFF"><a
href="ACWWTutorial0501123.doc"><strong><font
color="#000000">Download
  the latest ACWW tutorial
here.</font></strong></a></font><font color="#000000"><a
href="ACWWTutorial0501123.doc"></a></font></h2>
</body>
</html>
```

C.7. Step2a.html

```
<html>
<head>
<title>ACWW: Step2a</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong>Step
                      2: Enter the Webpage's Links &amp;
Headings</strong></font></div></td>
                  <td width="135"><div align="center"><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div
align="center"><strong></strong></div></td>
                  <td width="135"> </td>
                </tr>
              </table>
            </div></td>
          <td width="1%"></td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
        <tr>
          <td width="1%" bgcolor="#0000CC"> </td>
```

```
          <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="ACWW.php">
              <blockquote>
                <p> </p>
                <p><b>1. Enter a Label (a unique
combination of number or letters)
                  for the WebPage</b>:</p>
                <p>
                  <input name="label" type="text" id="name"
value=""  size="80" maxlength="150">
                </p>
                <p><strong>2. Enter the Heading Labels
(seperated by a blank line)</strong><strong>:</strong></p>
                <p>
                  <textarea name="headings" cols="100"
rows="20"></textarea>
                </p>
                <p align="center">
                  <input name="add" type="submit" id="next"
value="Add Links to the Headings">
                </p>
              </blockquote>
            </form></td>
          <td width="1%" bgcolor="#0000CC"> </td>
        </tr>
      </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
</body>
</html>
```

C.8. Step2b_1button.html

```
                  <input name="another" type="submit"
id="another" value="Enter Link Labels for the next
Heading">
                </p>
```

```
              </blockquote>
            </form></td>
          <td width="1%" bgcolor="#0000CC"> </td>
        </tr>
      </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
</body>
</html>
```

C.9. Step2b_2button.html

```
                  <input name="another" type="submit"
id="another" value="Enter the Headings and Links of Another
Webpage">    
                   <input name="next" type="submit"
id="next" value="Next Step">
                </p>
              </blockquote>
            </form></td>
          <td width="1%" bgcolor="#0000CC"> </td>
        </tr>
      </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
</body>
</html>
```

C.10. Step2b_body.html

```
"</strong></font><br>
                    (Separate each new item with a blank
line.)</p>
                  <p align=center><b>
                    <textarea name="links" cols="100"
rows="15"></textarea>
                    </b></p>
                  <p align="center">
```

C.11. Step2b_header.html

```
<html>
<head>
<title>ACWW: Step2b</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong>Step
                      2: Enter the Webpage's Links &amp;
Headings</strong></font></div></td>
```

```
                    <td width="135"><div align="center"><font
color="#FFFFFF"></font></div></td>
                    <td width="135"><div
align="center"><strong></strong></div></td>
                    <td width="135"> </td>
                  </tr>
                </table>
              </div></td>
            <td width="1%"></td>
          </tr>
        </table>
      </td>
    </tr>
    <tr>
      <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
          <tr>
            <td width="1%" bgcolor="#0000CC"> </td>
            <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="ACWW.php">
                <blockquote>
                  <p> </p>
                  <p align="center"><font
size="+2"><strong>Enter the Link Labels
                    for "
```

C.12. Step3_footer.html

```
                      </table></td>
                  </tr>
                </table>
                <p>  </p>
                <blockquote>
<p align="center"><input name="next" type="submit"
id="next" value="Next Step"></p>
                </blockquote>
              </form></td>
            <td width="1%" bgcolor="#0000CC"> </td>
          </tr>
        </table></td>
    </tr>
    <tr>
      <td bgcolor="#0000CC"> </td>
```

```
  </tr>
</table>
</body>
</html>
```

C.13. Step3_header.html

```
<html>
<head>
<title>ACWW: Step3</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong><font color="#FFFFFF">Step
                      2: Enter the Webpage's Links &amp;
Headings</font></strong></font><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong>Step
                      3: Group Goals &amp;
Webpages</strong></font></div></td>
```

```
                    <td width="135"><div
align="center"><strong></strong></div></td>
                    <td width="135"> </td>
                  </tr>
                </table>
              </div></td>
            <td width="1%"></td>
          </tr>
        </table>
      </td>
    </tr>
    <tr>
      <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
          <tr>
            <td width="1%" bgcolor="#0000CC"> </td>
            <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="">
                <p> </p>
                <br><center>
                  <p><strong><font size="+1">Select all the
Webpages that the user
                    would follow to accomplish each
goal.</font></strong></p>
                  <br>
                  </center>
                <table width="100%" border="1"
cellspacing="1" cellpadding="1">
                  <tr>
                    <td width="4%"> </td>
                    <td width="96%"><div
align="center"><strong>WebPages</strong></div></td>
                  </tr>
                  <tr>
                    <td><div align="center"><strong>Goals<br>
                        </strong></div></td>
                    <td><table width="100%" border="2"
cellpadding="1" cellspacing="1" bordercolor="#000000">
```

C.14. Step4.html

```
<html>
<head>
<title>ACWW: Step4</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong><font color="#FFFFFF">Step
                      2: Enter the Webpage's Links &amp;
Headings</font></strong></font><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div align="center"><font
color="#FFFFFF"><strong>Step
                      3: Group Goals and
WebPages</strong></font></div></td>
                  <td width="135"><div
align="center"><strong><font color="#FF0000">Step
                      4: Identify Heading &amp;
Links</font></strong></div></td>
                  <td width="135"><strong></strong></td>
                </tr>
              </table>
            </div></td>
          <td width="1%"></td>
        </tr>
      </table>
```

```
    </td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
        <tr>
          <td width="1%" bgcolor="#0000CC"> </td>
          <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="">
              <p>  </p>
              <blockquote>
                <p align="center"><strong>Goal 1 compared
to Webpage 1</strong></p>
                <p align="center"><br>
                  Goal 1: Audiometer, instrument for
testing hearing. The audiometer
                  is an essentially simple instrument that
produces pure tones
                  of various fixed pitches (frequencies)
heard through headphones.
                  Hearing is tested one ear at a time. The
operator can switch
                  between frequencies and repeat the
process with each frequency.
                  Typically, sensitivity may be tested at
frequencies of 125 hertz
                  (Hz, or cycles per second), 250 Hz, 500
Hz, 1000 Hz, 2000 Hz,
                  4000 Hz, 8000 Hz, and 12,000 Hz. As an
alternative to testing
                  the normal mode of hearing through
headphones, hearing by bone
                  conduction can be tested. Hearing is
never uniform over all
                  frequencies and commonly varies widely at
different frequencies.
                  Internally, audiometers consist of a
transistorized, variable-frequency
                  audio oscillator-usually a simple
feedback device-capable of
                  producing a sinusoidal (near sine-wave)
output</p>
                <p align="left">.<br>
                  <strong>Check one or more
&#8220;correct&#8221; links for accomplishing
                  Goal 1 on Webpage 1:</strong><br>
```

```
                    <input type="checkbox" name="checkbox"
value="checkbox">
                    Performing Arts<br>
                    <input type="checkbox" name="checkbox2"
value="checkbox">
                    Music<br>
                    <input type="checkbox" name="checkbox3"
value="checkbox">
                    Musicians and Composers<br>
                    <input type="checkbox" name="checkbox4"
value="checkbox">
                    Musical Instruments<br>
                    <input type="checkbox" name="checkbox5"
value="checkbox">
                    Theater<br>
                    <input type="checkbox" name="checkbox6"
value="checkbox">
                    Dance<br>
                    <input type="checkbox" name="checkbox7"
value="checkbox">
                    Cinema, TV and Broadcasting<br>
                    <input type="checkbox" name="checkbox8"
value="checkbox">
                    Religion &amp;amp; Philosophy<br>
                    <input type="checkbox" name="checkbox9"
value="checkbox">
                    Philosophy<br>
                    <input type="checkbox" name="checkbox10"
value="checkbox">
                    Religion &amp;amp; Religious Groups<br>
                    <input type="checkbox" name="checkbox11"
value="checkbox">
                    Religious Figures<br>
                    <input type="checkbox" name="checkbox12"
value="checkbox">
                    Scripture<br>
                    <input type="checkbox" name="checkbox13"
value="checkbox">
                    Theology &amp;amp; Practices<br>
                    <input type="checkbox" name="checkbox14"
value="checkbox">
                    The Occult<br>
                    [Continue through rest of headings and
links as entered by the
                    researcher for that webpage] </p>
```

C.15. Step4_footer.html

```
<p align="center"><input name="next" type="submit"
id="next" value="Next"></p>
              </blockquote></form></td><td width="1%"
bgcolor="#0000CC"> </td>
          </tr></table></td></tr><tr><td
bgcolor="#0000CC"> </td></tr></table></body></html>
```

C.16. Step4_header.html

```
<html>
<head>
<title>ACWW: Step4</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                  <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                        1: Enter Goal
Statment(s)</font></strong></div></td>
```

```
                    <td width="135"><div align="center"><font
color="#FF0000"><strong><font color="#FFFFFF">Step
                        2: Enter the Webpage's Links &amp;
Headings</font></strong></font><font
color="#FFFFFF"></font></div></td>
                    <td width="135"><div align="center"><font
color="#FFFFFF"><strong>Step
                        3: Group Goals and
WebPages</strong></font></div></td>
                    <td width="135"><div
align="center"><strong><font color="#FF0000">Step
                        4: Identify Heading &amp;
Links</font></strong></div></td>
                    <td width="135"><strong></strong></td>
                  </tr>
                </table>
              </div></td>
            <td width="1%"></td>
          </tr>
        </table>
      </td>
    </tr>
    <tr>
      <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
          <tr>
            <td width="1%" bgcolor="#0000CC"> </td>
            <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="">
                <p>  </p>
                <blockquote>
```

C.17. Step5.html

```
<html>
<head>
<title>ACWW: Step5</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
```

```
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                 <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong><font color="#FFFFFF">Step
                      2: Enter the Webpage's Links &amp;
Headings</font></strong></font><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div align="center"><font
color="#FFFFFF"><strong>Step
                      3: Group Goals and
WebPages</strong></font></div></td>
                  <td width="135"><div
align="center"><strong><font color="#FFFFFF">Step
                      4: Identify Heading &amp;
Links</font></strong></div></td>
                  <td width="135"><strong><font
color="#FF0000">Step 5: Enter
                        Options </font></strong></td>
                </tr>
              </table>
            </div></td>
          <td width="1%"></td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
        <tr>
          <td width="1%" bgcolor="#0000CC"> </td>
          <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="">
```

```
                <p>  </p>
                <blockquote>
                  <p><strong>1. Select Topic Space:<br>
                    <select name="space" id="space">
                      <option value="French-Monde">Francais-
Monde (300 factors)
                      </option>
                      <option
value="tasa03">General_Reading_up_to_03rd_Grade (300
                      factors) </option>
                      <option
value="tasa06">General_Reading_up_to_06th_Grade (300
                      factors) </option>
                      <option
value="tasa09">General_Reading_up_to_09th_Grade (300
                      factors) </option>
                      <option
value="tasa12">General_Reading_up_to_12th_Grade (300
                      factors) </option>
                      <option value="tasaALL"
selected>General_Reading_up_to_1st_year_college
                      (300 factors) </option>
                      <option value="adamMedical">Small
Medical Encylopedia</option>
                    </select>
                    </strong></p>
                  <p><strong>3. Link Elaboration
Options:</strong><br>
                    <input type="radio"
name="link_elaboration" value="no">
                    NO link elaboration (use just the link
test printed on the webpage)<br>
                      <input name="link_elaboration"
type="radio" value="full" checked >
                    FULL link elaboration<br>
                           Minimum
word frequency of  
                    <input name="link_frequency" type="text"
id="link_frequency" value="50" size="10" maxlength="10">
                      <br>
                         Minimum
cosine value of
</strong>      
                    <input name="link_cosine" type="text"
id="link_cosine" value=".50" size="10" maxlength="10">
                  </p>
```

```
              <p><strong>4. Heading Elaboration
Options:</strong><br>
                <input type="radio"
name="heading_elaboration" value="no">
                  NO heading elaboration (use just the text
printed on the webpage)<br>
                <input type="radio"
name="heading_elaboration" value="minimum">
                MINIMAL elaboration<br>
                <input name="heading_elaboration"
type="radio" value="full" checked>
                  FULL heading elaboration<br>
                     Minimum word
frequency of  
                <input name="heading_frequency" type="text"
id="heading_frequency" value="50" size="10" maxlength="10">
                <br>
                     Minimum
cosine value of
</strong>      
                <input name="heading_cosine" type="text"
id="heading_cosine" value=".50" size="10" maxlength="10">
              </p>

              <p><strong>5. Enter the Predicted Mean Total
Clicks Formula:</strong><br>
                     Predicted
mean total clicks = <br>

         
                  <input name="base" type="text"
value="2.292" size="20" maxlength="20">
                  <br>
                      
                    +
                  <input name="unfamiliar" type="text"
value="1.757" size="20" maxlength="20">
                if correct link is unfamiliar<br>
                      
                    +
                  <input name="weakscent" type="text"
value="1.516" size="20" maxlength="20">
                if correct link has a weak-scent<br>
                      
                    +
```

```
                    <input name="clcompetingheadings"
type="text" value="0.655" size="20" maxlength="20">
                    times the number of competing links
nested under competing heading(s)<br>
                        
                    +
                    <input name="clcorrectheadings"
type="text" value="0" size="20" maxlength="20">
                    times the number of competing links
nested under correct heading(s)<br>
                           
                     +
                    <input name="competingheadings"
type="text" value="0" size="20" maxlength="20">
                    times the number of competing headings
</p>

                <p align="center"> <font
size="+2"></font></p>
                     <p align="center">
                     <input name="another" type="submit"
id="another" value="Enter another set of options to run
with the Goals/Webpages entered">    
                    <input name="next" type="submit"
id="next" value="Next Step">
                     </p>
                </blockquote>
              </form></td>
            <td width="1%" bgcolor="#0000CC"> </td>
          </tr>
        </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
</body>
</html>
```

C.18. Step6.html

```
<html>
<head>
<title>ACWW: Step6</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
</head>
<body bgcolor="#0000cc">
<table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
  <tr>
    <td height="8" bgcolor="#0000CC">
     <table width="100%" height="100%" cellpadding="0"
cellspacing="0" border="0">
        <tr>
          <td width="1%" height="34"></td>
          <td width="99%"><div align="center">
              <table width="100%" border="1"
cellpadding="1" cellspacing="1" bordercolor="#0000CC">
                <tr>
                 <td width="135"> <div
align="center"><strong><font color="#FFFFFF">Step
                      1: Enter Goal
Statment(s)</font></strong></div></td>
                  <td width="135"><div align="center"><font
color="#FF0000"><strong><font color="#FFFFFF">Step
                      2: Enter the Webpage's Links &amp;
Headings</font></strong></font><font
color="#FFFFFF"></font></div></td>
                  <td width="135"><div align="center"><font
color="#FFFFFF"><strong>Step
                      3: Group Goals and
WebPages</strong></font></div></td>
                  <td width="135"><div
align="center"><strong><font color="#FFFFFF">Step
                      4: Identify Heading &amp;
Links</font></strong></div></td>
                  <td width="135"><strong><font
color="#FF0000">Step 5: Enter
                    Options </font></strong></td>
                </tr>
              </table>
            </div></td>
          <td width="1%"></td>
```

```
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td><table width="100%" border="0" cellpadding="0"
cellspacing="0" bordercolor="#0000CC">
        <tr>
          <td width="1%" bgcolor="#0000CC"> </td>
          <td width="98%" bgcolor="#FFFFFF"><form
name="form1" method="post" action="">
              <p>  </p>
              <blockquote>
                <p><strong>1. Enter an email address to
send the results:<br>

</strong>        &n
bsp;
                  <input name="email_addy" type="text" "
size="50" maxlength="200">
                </p>
                <p align="center"> <font
size="+2"></font></p>
                  <p align="center">
    
                  <input name="finish" type="submit"
id="finish" value="Save email address and submit all data
into the queue.  This may take a moment.">
                  </p>
              </blockquote>
            </form></td>
          <td width="1%" bgcolor="#0000CC"> </td>
        </tr>
      </table></td>
  </tr>
  <tr>
    <td bgcolor="#0000CC"> </td>
  </tr>
</table>
</body>
</html>
```

C.19. acww-termVectors.cgi

```
#!/usr/bin/perl
use strict;
#use CGI qw (:standard);
use CGI qw ( -debug );
use lib "/usr/local/apache/cgi-bin";
use diptiscripts;
$| = 1;

my $q = new CGI;
my ($link, @links, $filename, $cokid);
my @textArray = ();
my ($returned, $result);
my $end = 0;
my $flag = 1;
my $y;
my $x;
my %termVectors;
my $LSAIdx;
my $space;
my $tempdir = '/usr2/home/brownr/temp/';

#subroutine to call the LSA functions...Jose's scripts
sub call {

      my ($verbose, $corporaDir, $LSAFactors, $LSAbindir,
$LSAIdx, $bothCosinesAndVectors);
      my ($flag, $LSAIdx, $textString) = @_;
      my $vector = 0;

      $verbose = 0;
      $corporaDir = '/usr/lsa/spaces/';
      $LSAFactors = "300";
      $LSAbindir = '/usr/lsa/bin/';

      $vector = getVector ($verbose, $corporaDir,
$LSAFactors, $LSAbindir, $LSAIdx, $flag, $textString);

      my $singular = 's';

      my $vectorLength = getVectorLength ($corporaDir,
$LSAFactors, $LSAbindir, $LSAIdx, $vector, $singular);

     return $vectorLength;
```

```
}


#################################################################
#
#MAIN
#################################################################
#
#$cokid = $q->cookie('Test_Cookie');
#if (!$cokid) {exit;}
#$cokid = s/^\s//g; $cokid = s/\s$//g;
srand(time()^($$ + ($$ <<15)));
$cokid = "acwwt" . rand(1000);
$filename = $tempdir . $cokid . "8.txt";
open LINES, ">$filename" or die "Can't open the text file";


$space = $q->param('Space');
$space =~ s/^\s+//g; $space =~ s/\s+$//g;
$link = $q->param('Links');
$link =~ s/^\s+//g; $link =~ s/\s+$//g;
@links = split (/\r/, $link);
foreach (@links) {
     $_ =~ s/^[\W\s]//g;
     }
foreach (@links) {
     if ($_ =~ /^\w+/) {
          print LINES ($_, "\n");
           chomp;
                $textArray[$end] = $_;
           $end++;
     }
}

close LINES;


#if ($space eq "General_Reading_Up_to_3rd_Grade") { $LSAIdx
= "tasa03";}
#if ($space eq "General_Reading_Up_to_6th_Grade") {$LSAIdx
= "tasa06";}
#if ($space eq "General_Reading_Up_to_9th_Grade") {$LSAIdx
= "tasa09";}
#if ($space eq "General_Reading_Up_to_12th_Grade") {$LSAIdx
= "tasa12";}
```

```
#if ($space eq "General_Reading_Up_to_1st_Year_College")
{$LSAIdx = "tasaALL";}
#if ($space eq "French-Monde-1993") {$LSAIdx = "French-
Monde";}
$LSAIdx = $space;

$filename = "";
$filename = $tempdir . $cokid . "8.txt";
open (LINES, "<$filename") or die "Can't open the bloddy
text file";
$filename = "";
$filename = $tempdir . $cokid . "9.txt";
open (ANS, ">$filename") or die "Can't open all the text
files";


if ($space eq "French-Monde-1993") {
     for ($y = 0; $y< @textArray; $y++)
     {
          $textArray[$y] = translate($textArray[$y],
"English");
     }
}


for ($y = 0; $y< @textArray; $y++)
{

        $returned = call ($flag, $LSAIdx, $textArray[$y]);
     $result = sprintf ("%.2f", $returned);
     if ($space eq "French-Monde-1993") {
          $termVectors{$result} = translate
($textArray[$y], "French");
     }
     else { $termVectors{$result} = $textArray[$y];}
        print ANS ("The term vector length of
$textArray[$y] = ", "$returned\n");


}

close LINES;
close ANS;

foreach $y (reverse sort keys %termVectors)
{
#        print  ("$termVectors{$y}\n");
```

```
     if ($y < 0.80) {$x = $x + 2};
     print ( $y );
}
$x = $x + 4;

foreach $y (sort keys %termVectors)
{
     if ($y < 0.80) {
#    print "$termVectors{$y}\n\n";
}
}

my $rmdir = $tempdir . $cokid . "*";
`rm $rmdir`;
```

C.20. acww-one2many.cgi

```
#!/usr/bin/perl
use strict;
#use CGI qw (:standard);
use CGI qw (-debug);
use lib "/usr/local/apache/cgi-bin";
use diptiscripts;
$| = 1;

my $q = new CGI;
my ($cokid, $filename, $finalgoal, $link, , $goal ,@links,
$frequency, $cosine, %words, $c, $space, $LSAIdx);
my @textArray = ();
my $end = 0;
my $f;
my $returned;
my $tempdir = '/usr2/home/brownr/temp/';

#this part calculates the one to many comparison

sub form_textfile {
my ($end1, $LSAIdx, $targetString, $textString) = @_;

my ($flag, $verbose, $corporaDir, $LSAFactors, $LSAbindir,
$bothCosinesAndVectors, $cosines);
my $vector = 0;
$verbose = 0;
$corporaDir = '/usr/lsa/spaces/';
```

```
$LSAFactors = "";
$LSAbindir = '/usr/lsa/bin';
$flag = 0;

$cosines = one2many ($verbose, $corporaDir, $LSAFactors,
$LSAbindir, $LSAIdx, $flag, $targetString, $textString);
return $cosines;
}

#give the frequency taken from the form and the text as
input
#output is a hash

####################################
##MAIN
####################################
#$cokid = $q->cookie('Test_Cookie');
#if (!$cokid) {exit;}
#$cokid = s/^\s//g; $cokid = s/\s$//g;
srand(time() ^($$ + ($$ <<15))) ;
$cokid = "acww" . rand(10000);
$filename = $tempdir . $cokid . "11.txt";
open (FOO, ">$filename") || die "Cant open the input file";
$filename = "";
$filename = $tempdir  . $cokid . "12.txt";
open (INTER, ">$filename") || die "Cant open the
intermediate file.";

$finalgoal = $q->param('Goal');
$finalgoal =~ s/^\s+//g; $finalgoal =~ s/\s+$//g;
$space = $q->param('Space');
$space =~ s/^\s+//g; $finalgoal =~ s/\s+$//g;

print FOO ("$finalgoal", "\n");
print FOO ("\n");
$link = $q->param('Links');
$link =~ s/^\s+//g; $link =~ s/\s+$//g;
@links = split (/\r/, $link);
foreach (@links) {
     $_ =~ s/[\W]/ /g;
}
foreach (@links) {
     if ($_ =~ /^\W*\w+/) {
          print FOO ("$_ "); }
     else  {
          print FOO ("\n\n");}
}
```

```
close FOO;

#if ($space eq "General_Reading_Up_to_3rd_Grade") { $LSAIdx
= "tasa03";}
#if ($space eq "General_Reading_Up_to_6th_Grade") {$LSAIdx
= "tasa06";}
#if ($space eq "General_Reading_Up_to_9th_Grade") {$LSAIdx
= "tasa09";}
#if ($space eq "General_Reading_Up_to_12th_Grade") {$LSAIdx
= "tasa12";}
#if ($space eq "General_Reading_Up_to_1st_Year_College")
{$LSAIdx = "tasaALL";}
#if ($space eq "French-Monde-1993") {$LSAIdx = "French-
Monde";}
$LSAIdx = $space;

$filename = "";
$filename =  $tempdir . $cokid ."11.txt";
open (INP, "<$filename") || die "cant open the input
file.";
my $y;
my @input = ();
$y = 0;
while (<INP>)
{
        if ($_ =~ /^\W*\w+/)
        {
          chomp;
       $input[$y] = $input[$y] . $_;
     }
     else {
       $y++;
     }
     if ($space eq "French-Monde-1993") {
          $input[$y] = translate($input[$y], "English");
     }
}

if ($space eq "French-Monde-1993")
     { $goal = translate($input[0], "French");}
else {$goal = $input[0];}
for ($y = 1; $y< @input; $y++)
{
        $returned = form_textfile ($end, $LSAIdx,
$input[0], $input[$y]);
     $returned = sprintf ("%.2f", $returned);
```

```
     if ($space eq "French-Monde-1993") { $input[$y] =
translate($input[$y],"French");}
     print $returned;
     print INTER ("The cosine between goal and $input[$y] =
", "$returned\n");
}

close INTER;
my $rmdir = $tempdir . $cokid . "*";
`rm $rmdir`;
```

C.21. acww-elaborate.cgi

```
#!/usr/bin/perl
# /nph-elaborate1.cgi Space=tasaALL Frequency=50 Cosine=.50
Links=Music
use strict;
#use CGI qw (:standard);
use CGI qw ( -debug );
use lib "/usr/local/apache/cgi-bin";
use diptiscripts;
$| = 1;

my $q = new CGI;
my ($d, $word, $finalgoal, $link, @links, $cokid,$filename,
$frequency, $cosine, %words, $c, $space, $LSAIdx);
my @textArray = ();
my $end = 0;
my $f;
my $returned;
my $corporaDir = '/usr/lsa/spaces';
my $LSAFactors = "";
my $LSAbindir = '/usr/lsa/bin';
my $tempdir = '/usr2/home/brownr/temp/';

#this part calculates the one to many comparison

sub form_textfile {
my ($end1, $LSAIdx, $targetString, $textString) = @_;

my ($flag, $verbose, $corporaDir, $LSAFactors, $LSAbindir,
$bothCosinesAndVectors, $cosines);
my $vector = 0;
```

```
$verbose = 0;
#$corporaDir = '/usr/lsa/spaces/';
#$LSAFactors = "";
#$LSAbindir = '/usr/lsa/bin';
$flag = 0;


$cosines = one2many ($verbose, $corporaDir, $LSAFactors,
$LSAbindir, $LSAIdx, $flag, $targetString, $textString);
#print ("%cosines", "\n");
return $cosines;
}

#give the frequency taken from the form and the text as
input
#output is a hash
sub nearest {

my ($LSAIdx, $cosine, $wordFreq, $textString) = @_;
my $verbose = 0;
#my $corporaDir = '/usr/lsa/spaces/';
#my $LSAFactors = "";
#my $LSAbindir = '/usr/lsa/bin';
my $bothCosinesAndVectors = 1;
my $flag = 0;
my $Mparameter = 500;
my ($i, %wordCosines);
%wordCosines = nearestNeighbors($verbose, $corporaDir,
$LSAFactors, $LSAbindir, $LSAIdx, $flag,$Mparameter,
$textString, $wordFreq, $bothCosinesAndVectors, $cosine);
return %wordCosines;
}


####################################
##MAIN
####################################

#$cokid = $q->cookie('Test_Cookie');
#if (!$cokid) {exit;}
#$cokid =~ s/^\s//g; $cokid =~ s/\s$//g;
srand(time() ^($$ + ($$ <<15))) ;
$cokid = "acww" . rand(10000);
$filename = $tempdir . $cokid . "5.txt";
open (FOO, ">$filename") || die "Cant open the input file";
$filename = "";
$filename = $tempdir . $cokid . "6.txt";
```

```
open (INTER, ">$filename") || die "Cant open the
intermediate file.";

$space = $q->param('Space');
$space =~ s/^\s+//g; $space =~ s/\s+$//g;

$link = $q->param('Links');
$link =~ s/^\s+//g; $link =~ s/\s+$//g;
@links = split (/\r/, $link);
foreach (@links) {
     $_ =~ s/^[\W\s]//g;
     $_ =~ tr/A-Z/a-z/;
     }
foreach (@links) {
     if ($_ =~ /^\W*\w+/) {
          print FOO ("$_ "); }
     else {
          print FOO ("\n\n"); }
}
$frequency = $q->param('Frequency');
$cosine = $q->param('Cosine');
#This is the calculating the near neighbors part
close FOO;

#if ($space eq "General_Reading_Up_to_3rd_Grade") { $LSAIdx
= "tasa03";}
#if ($space eq "General_Reading_Up_to_6th_Grade") {$LSAIdx
= "tasa06";}
#if ($space eq "General_Reading_Up_to_9th_Grade") {$LSAIdx
= "tasa09";}
#if ($space eq "General_Reading_Up_to_12th_Grade") {$LSAIdx
= "tasa12";}
#if ($space eq "General_Reading_Up_to_1st_Year_College")
{$LSAIdx = "tasaALL";}
#if ($space eq "French-Monde-1993") {$LSAIdx = "French-
Monde";}
$LSAIdx = $space;
my $freqFile = `ls $corporaDir/$LSAIdx/*.freq`; chomp
$freqFile;

$filename = "";
$filename = $tempdir . $cokid . "5.txt";
open (INP, "<$filename") || die "cant open the input
file.";
my $y;
my @input = ();
$y = 0;
```

```
while (<INP>)
{
        if ($_ =~ /^\W*\w+/)
        {
            chomp;
                $input[$y] = $input[$y] . $_;
     }
     else {
                $y = $y + 1;
           }
}

for ($y =0; $y <= @input; $y = $y +1)  {

     if ($input[$y] =~ /^\W*\w+/)
          {

            if ($space eq "French-Monde-1993")
          {
               $word = translate($input[$y], "English");
          }
          else {
               $word = $input[$y];
          }
               print INTER ($word, " ");
          my @wordlist = split(/ /,$word);
          foreach (@wordlist) {
               $_ =~ s/^\s//g;}
          my $numbers = @wordlist;
          if ($numbers == 1) {
               my %whole = getFreq($LSAIdx, $corporaDir,
$freqFile, @wordlist);
               my @c = keys %whole;
               $d = $whole{$c[0]};
               }
          else { $d = 1;}
          if ($d != 0) {
                    if ($space eq "French-Monde-1993") {
                         %words = nearest ($LSAIdx,
$cosine, $frequency, $word);}
                    else {
                     %words = nearest ($LSAIdx, $cosine,
$frequency, $word);
               }
                    foreach $c (reverse sort keys %words)
                    {
                    print INTER ($words{$c}, " ");
```

```
                }
            }
                print INTER "\n\n";

        }
}



close INTER;

$filename = "";
$filename = $tempdir . $cokid . "6.txt";
open (LINES, "<$filename") or die "Can't open the bloddy
text file";
$filename = "";
$filename = $tempdir . $cokid . "7.txt";
open (ANS, ">$filename") or die "Can't open $filename";



while (<LINES>)
{
        if ($_ =~ /^\W*\w+/)
        {
           chomp;
           $textArray[$end] = $textArray[$end] . $_;
         }
       else
        {
        $end = $end + 1;
        }

}
close LINES;

for ($y = 0; $y< @textArray; $y++)
{
     if ($space eq "French-Monde-1993") {
          $word = translate ($textArray[$y], "French");
     }
     else {
          $word = $textArray[$y];
     }
        print  ("$word", "\n");
     print ANS $word, "\n";
```

```
}
close ANS;

my $rmdir = $tempdir . $cokid . "*";
`rm $rmdir`;
```

C.22. create_query.sql

```
CREATE TABLE `correct_heading` (
  `correct_headi_num` int(11) NOT NULL auto_increment,
  `matrix_num` int(11) default '0',
  `head_num` int(11) default '0',
  `job_num` int(11) default '0',
  PRIMARY KEY  (`correct_headi_num`),
  KEY `correct_heading_num` (`correct_headi_num`),
  KEY `head_key` (`head_num`),
  KEY `headingcorrect_heading` (`head_num`),
  KEY `job_num` (`job_num`),
  KEY `matrix_num` (`matrix_num`),
  KEY `matrixcorrect_heading` (`matrix_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `correct_link` (
  `correct_link_num` int(11) NOT NULL auto_increment,
  `correct_head_num` int(11) default '0',
  `link_num` int(11) default '0',
  `job_num` int(11) default '0',
  PRIMARY KEY  (`correct_link_num`),
  KEY `correct_head_num` (`correct_head_num`),
  KEY `correct_headingcorrect_link` (`correct_head_num`),
  KEY `correct_link_num` (`correct_link_num`),
  KEY `job_num` (`job_num`),
  KEY `link_num` (`link_num`),
  KEY `Linkcorrect_link` (`link_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `goal` (
  `goal_num` int(11) NOT NULL auto_increment,
  `goal_label` longtext NOT NULL,
```

```
  `goal_statement` longtext NOT NULL,
  `goal_filename` longtext NOT NULL,
  `job_num` int(11) NOT NULL default '0',
  PRIMARY KEY  (`goal_num`),
  KEY `goal_num` (`goal_num`),
  KEY `job_num` (`job_num`),
  KEY `job_optionsGoal` (`job_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `heading` (
  `heading_num` int(11) NOT NULL auto_increment,
  `heading_label` longtext NOT NULL,
  `web_num` int(11) NOT NULL default '0',
  `elaborated_label` longtext,
  `job_num` int(11) default '0',
  PRIMARY KEY  (`heading_num`),
  KEY `heading_num` (`heading_num`),
  KEY `job_num` (`job_num`),
  KEY `web_num` (`web_num`),
  KEY `webpageheading` (`web_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `job_options` (
  `job_num` int(11) NOT NULL auto_increment,
  `job_space` varchar(50) NOT NULL default '',
  `job_link_freq` int(11) NOT NULL default '0',
  `job_link_cosine` int(11) NOT NULL default '0',
  `job_head_freq` int(11) NOT NULL default '0',
  `job_head_cosine` int(11) NOT NULL default '0',
  `job_email` varchar(255) NOT NULL default '',
  `job_id` longtext NOT NULL,
  `base` double NOT NULL default '0',
  `unfamiliar` double NOT NULL default '0',
  `weakscent` double NOT NULL default '0',
  `competinglinksundercompetingheadings` double NOT NULL
default '0',
  `competinglinksundercorrectheadings` double NOT NULL
default '0',
  `competingheadings` double default '0',
  PRIMARY KEY  (`job_num`),
  UNIQUE KEY `job_id` (`job_id`(255)),
  KEY `job_num` (`job_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `link` (
  `link_num` int(11) NOT NULL auto_increment,
  `link_label` longtext NOT NULL,
  `head_num` int(11) NOT NULL default '0',
  `elaborated_label` longtext,
  `job_num` int(11) default '0',
  PRIMARY KEY  (`link_num`),
  KEY `heading_num` (`head_num`),
  KEY `headingLink` (`head_num`),
  KEY `job_num` (`job_num`),
  KEY `link_num` (`link_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `matrix` (
  `matrix_num` int(11) NOT NULL auto_increment,
  `web_num` int(11) default '0',
  `job_num` int(11) default '0',
  `goal_num` int(11) default '0',
  PRIMARY KEY  (`matrix_num`),
  KEY `goal_num` (`goal_num`),
  KEY `Goalmatrix` (`goal_num`),
  KEY `head_num` (`web_num`),
  KEY `job_num` (`job_num`),
  KEY `job_optionsmatrix` (`job_num`),
  KEY `matrix_num` (`matrix_num`),
  KEY `webpagematrix` (`web_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `one_to_many_results` (
  `result_num` int(11) NOT NULL auto_increment,
  `matrix_num` int(11) NOT NULL default '0',
  `original_label` longtext,
  `elaborated_label` longtext NOT NULL,
  `cosine` double NOT NULL default '0',
  `heading_or_link` varchar(50) NOT NULL default '',
  `specific_heading` longtext NOT NULL,
  `heading_num` int(11) default '0',
  `correct` longtext,
  `job_num` int(11) default '0',
  PRIMARY KEY  (`result_num`),
  KEY `heading_num` (`heading_num`),
  KEY `job_num` (`job_num`),
  KEY `matrix_num` (`matrix_num`),
  KEY `matrixone_to_many_results` (`matrix_num`),
  KEY `result_num` (`result_num`)
```

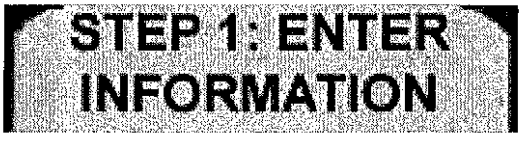
```
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


CREATE TABLE `webpage` (
  `web_num` int(11) NOT NULL auto_increment,
  `web_label` longtext NOT NULL,
  `job_num` int(11) default '0',
  PRIMARY KEY  (`web_num`),
  KEY `job_num` (`job_num`),
  KEY `job_optionswebpage` (`job_num`),
  KEY `web_num` (`web_num`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

C.23 Images

C.23.1. Step1a.jpg

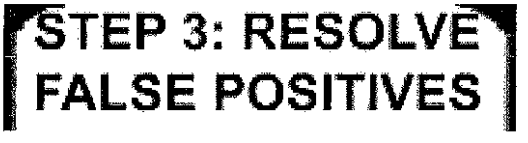
C.23.2. Step1u.jpg



C.23.3. Step2a.jpg

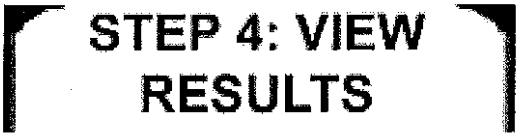
C.23.4. Step2u.jpg

C.23.5. Step3a.jpg

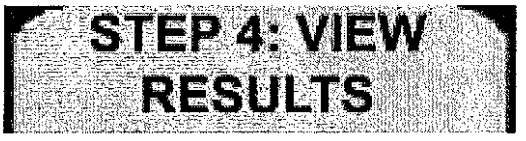


C.23.6. Step3u.jpg

C.23.7. Step4a.jpg



C.23.8. Step4u.jpg



REFERENCES

[Bias94]
Bias, R.G. and Mayhew, D. J., Cost-Justifying Usability, Academic Press, London, New York, 1994.

[Blackmon02]
Blackmon, M. H., Polson, P.G, Kitajima, M., & Lewis, C. "Cognitive Walkthrough for the Web," CHI 2002 Conference on Human Factors in Computing Systems, ACM Press, 2002, pp. 463-470.

[Blackmon03]
Blackmon, M.H., Kitajima, M., & Polson, P.G., "Web Usability: Repairing Usability Problems Identified by the Cognitive Walkthrough for the Web," CHI 2003 Conference on Human Factors in Computing Systems, ACM Press, 2003, pp. 497-504.

[Blackmon05]
Blackmon, M.H., Kitajima, M., & Polson, P.G., "Tool for Accurately Predicting Website Navigation Problems, Non-Problems, Problem Severity, and Effectiveness of Repairs," CHI 2005 Conference on Human Factors in Computing Systems, ACM Press, 2005.

[Byme99]
Byme, M. D., John, B. E., Wehrle, N. S., & Crow, D. C., "The Tangled Web we Wove: A Taskonomy of WWW Use," Proceedings of CH'99, ACM Press, 1999, pp 544-551.

[Chi00]
Chi, E., Pirolli, P., Chen, K., & Pitkow, J., "The Scent of a Site: A System for Analyzing and Predicting Information Scent, Usage, and Usability of a Web Site," Proceedings of Chi 2000, ACM Press, 2001, pp. 161-168.

[Chi01]
Chi, E., Pirolli, P., Chen, K., & Pitkow, J., "Using information scent to model user information needs and actions and the Web," Proceedings of CHI 2001, ACM Press. 2001, pp. 490-497.

[Chi03]
Chi, E.H., et al, "The Bloodhound Project: Automating Discovery of Web Usability Issues using the InfoScent™ Simulator," CHI 2003 Conference on Human Factors in Computing Systems, ACM Press, 2003, pp. 505-512.

[Hertzum03]
Hertzum, M. and Jacobsen, N., "The Evaluator Effect: A Chilling Fact About Usability Evaluation Methods," International Journal of Human-Computer Interaction, Vol 15, pp. 183-204.

[Huberman98]
Huberman, B.A., et al, "Strong Regularities in World Wide Web Surfing," Science, April 3, 1998, Vol. 280, num 5360, pp. 95-97.

[Kintsch98]
Kintsch, W., Comprehension: A Paradigm for Cognition, Cambridge University Press, 1998.

[Kitajima95]
Kitajima M. and Polson, P.G., "A Comprehension-Based Model of Correct Performance and Errors in Skilled Display-Based, Human-Computer Interaction," International Journal of Human-Computer Studies, Vol. 43, pp. 65-99.

[Kitajima97]
Kitajima, M. and Polson, P.G., "A Comprehension-Based Model of Exploration," Human-Computer Interaction, Vol. 12, 1997, pp. 345-389.

[Kitajima00]
Kitajima, M. Blackmon, M.H., Polson, P.G., "A Comprehension-based Model of Web Navigation and its application to Web Usability Analysis," People and Computers, Vol XIV, Springer, 2000, pp. 357-373.

[Kitajima05]
Kitajima, M. Blackmon, M.H., Polson, P.G., "Cognitive Architecture for Website Design and Usability Evaluation: Comprehension and Information Scent in Performing by Exploration," HCI International Conference 2005, 2005.

[Koehler99]
Koehler, Wallace, "Classifying Websites and Webpages: The Use of Metrics and URL Characteristics as Markers," *Journal of Librarianship and Information Studies* 31, 1 March 1999, pp. 21-31.

[Landauer97]
Landauer, T. K., and Dumain, S. T., "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge," Psychological Review, Vol 104, 1997, pp. 211-240.

[Mantei88]
Mantei, M.M. and Teory, Toby T. J., "Cost/benefit for incorporating human factors in the software lifecycle," ACM Communications, 1988, Vol. 34, pp. 428-439.

[Morkes97]
Morkes, J. and Nielson, J., "Concise, SCANNABLE, and Objective: How to Write for the Web," http://www.useit.com/papers/webwriting/writing.html. 1997.

[Pirolli99]
Pirolli, P. and Card, S., "Information Foraging," Psychological Review, Vol. 106, 1999, pp. 643-675.

[Pirolli03]
Pirolli, P. and Fu, W-T, "SNIF-ACT: A model of information foraging on the World Wide Web," Ninth International Conference on User Modeling, Johnstown, PA, 2003.

[Pitkow97]
Pitkow, J.E. and Kehoe, C.M., "GVU's WWW User Surveys," http://www.gvu.gatech.edu/user surveys, 1997.

[Tullis98]
Tullis, T.S., "A Method for Evaluating Web Page Design Concepts," Chi'98 Conference Summary, 1998, pp. 323-324.

[Vredenburg02]
Vredenburg, Karel, Isensee, Scott, & Righi, Carol, User-Centered Design, Prentice Hall PTR, 2002.

VITA

Richard A. Brown has a Bachelor of Science from the University of North Florida in Computer and Information Sciences, 2001 and expects to receive a Master of Science in Computer and Information Sciences from the University of North Florida, December, 2005. Dr. Arturo Sanchez of the University of North Florida is serving as Richard's thesis adviser. Richard is currently employed as a National Depot Sales and Operations Manager at Computer, Inc. and has been with the company for 8 years. Prior to that, Richard worked for 6 months as a computer technician at Jacksonville University.

Richard has on-going interests in computer human interfaces. Richard has extensive programming experience with C++, JAVA, and Visual Basic. Richard's academic work has included the use of these languages as well as SQL, UNIX shell scripting, PERL, and PHP.