

Improving an Image Annotation and Retrieval Agent using Commonsense and Personal Knowledge

by

Kimberly G. Waters

B.S. Computer Science
University of Maryland Baltimore County (UMBC), 1999

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF

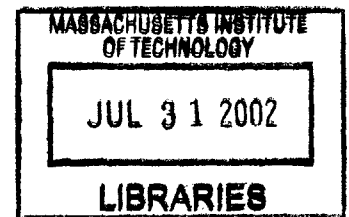
MASTER OF SCIENCE IN COMPUTER SCIENCE
AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2002

© 2002 MIT. All rights reserved.

The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic
copies of this thesis document in whole or in part.

BARKER



Signature of Author: _____
Department of Electrical Engineering and Computer Science
May 24, 2002

Certified by: _____
Henry Lieberman
Research Scientist, MIT Media Laboratory
sis Supervisor

Accepted by: _____
A.C. Smith
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Improving an Image Annotation and Retrieval Agent using Commonsense and Personal Knowledge

by

Kimberly G. Waters

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2002 in Partial Fulfillment of the
requirements for the Degree of Master of Science in
Computer Science

ABSTRACT

As digital photography continues to grow in popularity and usefulness, consumers can accumulate large quantities of images that may rarely be looked at over the course of a lifetime. Since computer vision has not yet become sophisticated enough to automatically recognize the content of images, the use of a software agent for integrated annotation and retrieval of images is proposed. This agent, ARIA, will combine commonsense knowledge, specific personal background information about the user, and the context in which they will use the images to assist him/her in the process of locating and using his/her images.

The act of composing an email or webpage provides the context needed for the agent to present relevant images. ARIA looks at the words closest to the cursor and uses those as a query to the photo database. Not only does ARIA look for direct keyword matches, but it also expands the query with related words to increase the likelihood of locating pictures. General relationships are garnered from Open Mind, a commonsense knowledge repository. Because consumer photography is very personal, a repository of information specific to the user is also included to bridge the gap.

Thesis Supervisor: Henry Lieberman
Title: Research Scientist, MIT Media Laboratory

Table of Contents

Abstract	3
1. Introduction	5
1.1 Research Problem	5
1.2 Motivation	7
1.3 Limitations and Assumptions	11
2. A User Scenario	12
3. Methodology	15
3.1. Knowledge Repositories	15
3.2 Query Expansion	24
3.3 Interface Changes	30
4. Related Work	34
4.1 Digital Image Retrieval	34
4.2 User Interface Agents	38
4.3 Knowledge Repositories	39
4.4 Knowledge-Based Applications	40
5. User Evaluation	43
6. Conclusion	44
7. Acknowledgements	44
References	45
Appendix A: Personal Repository Contents	47
Appendix B: Subset of Open Mind Contents	48
Appendix C: Query Expansion Trace for User Scenario	50

Introduction

1.1. Research Problem

1.1.1. ARIA Background

ARIA (Annotation and Retrieval Integration Agent) [Lieberman, Rosenzweig], a user interface agent, was created to combine the annotation, retrieval, and usage of digital images into a single application. This program allows users to accomplish two of the top three activities related to digital photography: emailing images and sharing photos with friends and family via web pages. ARIA endeavors to simplify the process of using digital photos, actively assisting users in their tasks rather than being an inflexible tool that passively waits to be acted upon.

The software agent sits in the user's email editor and as he/she types, the agent extracts keywords from the area closest to the text cursor. These keywords are used to query the image database and then rerank the images such that the most relevant photos are presented to the user at the top of the list. ARIA also helps the annotation process. Suppose there were recently taken pictures stored on your camera, and you wanted to use them. When you insert the flash card into the computer, ARIA automatically imports those pictures and places them at the top of your image retrieval window. If you choose to include a photo in your message, you simply drag it into the text. Any keywords in proximity to the insertion point would immediately be annotated to the photo to assist in future searches.

This type of interaction differs from traditional image retrieval applications in several ways. There is no need to:

- Open a separate image retrieval or viewing program
- Interrupt the creation of your message in order to perform a keyword search
- Respond to file dialog windows asking you to locate images or decide where in the file directory system to save images
- Instruct the system to import new images from the camera

- Cut and paste or upload images into an email account
- Manually annotate all images before using them (though ARIA does allow users to manually annotate images if they choose).

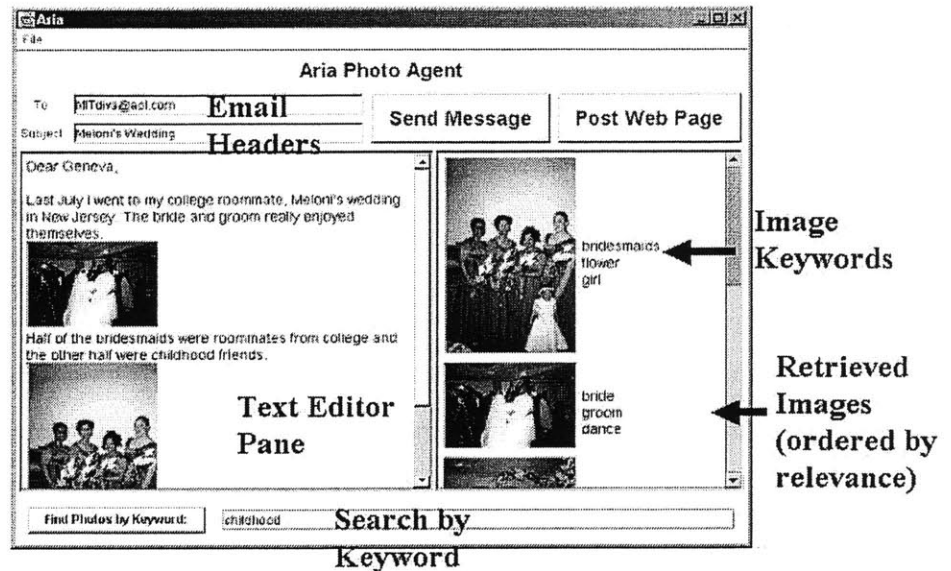


Figure 1.1: Original ARIA Interface

Agent technology is ideal in this situation because it utilizes potentially wasted CPU time, which would have been spent idly waiting for the user to do something. Instead, the agent continuously queries the image database for pictures that may be of interest to the user. Also, since the suggestions are unobtrusively displayed in the adjacent frame of the interface, there is minimal cost to the user if he/she glances at the results and deems them of no value at the moment. ARIA hides the complexity of incorporating photos into the text of an email message or web page and performs continuous image searches in order to assist the user. These two characteristics – providing low risk suggestions to the user and reducing complexity – are very important aspects of an agent's design [Rhodes, Maes].

1.1.2. Statement of Purpose

The aim of this phase of the project was to move ARIA beyond direct keyword matching to a more intelligent method of recommendation. The agent would be able to make better suggestions if it had additional knowledge about the world. This is known as the “knowledge-based approach” [14] and has been accomplished by incorporating commonsense knowledge into the query process.

This commonsense-enhanced ARIA combines general knowledge about the world (e.g. that dogs are pets) with specific personal information (e.g. that my dog’s name is Spot) in order to improve image retrieval. The original query is augmented by these related keywords. Query expansion enables the agent to locate images that previously would have been overlooked due to annotations that do not exactly match the word typed by the user.

ARIA also provides several options for increasing its knowledge base. The first is an interface that allows users to select from several activities that prompt users to “teach ARIA”. Another method of extracting information directly from the message permits users to highlight relevant sentences and instruct the agent to add the highlighted text to the personal repository of knowledge. The agent’s third technique for eliciting commonsense prompts the user for more information about the subject he/she was just writing about. When the user clicks the “Tell me More” button, customized questions are presented to the user.

1.2. Motivation

Consumer usage of digital cameras continues to rise as technologies improve in quality and decrease in price. These cameras, which premiered at \$35,000 and were only used by professionals can now be found for as low as \$99 with the average price being \$300-\$500 – about the same as a high quality film-shooting 35 millimeter SLR camera [Yaukey]. An estimated 5.5 million cameras were sold in the U.S. in 2001 with growth projected at approximately 31%, despite the recent economic downturn.

Also, since only 16% of U.S. households reported owning a digital camera, market saturation is not yet an issue [Keefe].

Why have these gadgets grown so popular? Consumers are enthralled by the idea of having instant pictures without sacrificing quality. Digital cameras eliminate the wait between taking pictures and viewing them. You can immediately see if a photo did not come out and retake it without incurring the additional expense of using and developing more film. With traditional cameras, before you could develop the pictures you had to wait until the roll of film had been used completely. Then you had to either drop the roll of film at you local film lab or mail it off to a company for developing. Often, by the time consumers got their pictures back, they had forgotten what event the pictures were from. Since pictures are frequently taken at one time events, it is impossible to retake a photo that did not come out well.

Unlike film photography, only 11.8% of digital images are ever printed. According to the Photo Marketing Association International [Longheier], the top three reasons cited by consumers for using digital cameras are:

- To send photos by email 77.3%
- To preserve memories 67.8%
- To share later with others 66.9%

Increased internet bandwidth makes it more practical to share images, and with new digital photo editing tools, users have the ability to create professional quality copies of their pictures (reducing red eye, cropping, rotating, and adjusting contrast and brightness). New computers capable of handling multimedia applications come preloaded with all of the image viewing and processing software a user could ask for, but these utilities often go unused due to their complexity. It can be fairly difficult for the user to get from the initial picture-taking event to the ultimate use of the images.

With traditional photography, people tend to accumulate pictures in shoeboxes [13] where they sit for years. It would be easier to locate and make use of these photos if they were catalogued and indexed, but that process can be very time consuming, and putting forth so much upfront effort on the off chance of improving the possibility of future usage is not very enticing to consumers. A comparable phenomenon occurs with digital images in that they sit in “virtual shoeboxes” (file folders on the computer) and are rarely, if ever, looked at. They have names that do not express the content of the images (e.g. P000139.jpg); this makes locating specific images a formidable task.

Similarly, computer users could improve their ability to utilize digital images if they were willing to invest the time to set up helpful infrastructure. The most basic methods of organization are to create separate folders for images based on their content or to rename the files to more accurately indicate the subject of the image. However, this option is less than optimal because when it is time to make use of a particular photo, the user is still required to wade through the file system trying to recall where they stored the image.

A promising area of research that could assist in locating photographs is content-based image retrieval [9]. This computer vision approach to the problem is attractive because it frees users from the hands-on task of manually organizing data for easier access. For example, you could simply inform the system that you were looking for a picture of a red car. This could be accomplished by selecting a similar image as an example of what you would like to locate or you could specify the characteristics of what you desire to find (a smooth, red object in the case of the car).

This type of query by example technique is helpful when a user wants to find an image similar to a sample image [6, 4]. Unfortunately, these methods only deal with low-level features such as color and texture. The average consumer typically thinks in terms of objects or people, not shape and texture, making it difficult for an untrained user to utilize the system.

Conventional keyword-based search and retrieval products provide another alternative. Image databases allow you to retrieve visual data based on associated metadata: semantic information about the image, such as the creation date, location, event, and identity of subjects. The metadata is linked to the image via user annotations. These systems also allow for browsing through lists or galleries of images [9]. Examples of this type of image database system include Portfolio [5], Kudo Image Browser [8], and Cumulus [3].

Until computer vision systems are able to look beyond color and texture to higher-level recognition of people, places, and objects, keyword-based image retrieval seems the most intuitive way for users to find the pictures they want. Of course, the first issue that comes to mind when annotations are mentioned is who will create them? Corporations can afford to hire people to manually create necessary metadata, but without the financial incentive, consumers are not attracted to the prospect of vast quantities of data entry. The upfront work required of users forms a barrier to use by all but the most dedicated of users. As a result, many opportunities to utilize images are missed. How can users detect and capitalize on these previously missed opportunities?

Developers often consider the task of using an image as being independent of the annotation and retrieval process [13]. The burden is placed on the user to figure out which applications are required to accomplish a task. He/she then needs to explicitly convert and transfer the necessary data from one application to another as necessary.

As stated previously, people mainly take pictures to document events and to tell stories about them. Imagine you are sending an email to a friend to tell him/her about a wedding you recently attended. Including pictures that you took at the event as well as related pictures from other events could enhance your story. What steps are necessary to get from the capturing of the images to the sending of the email message?

- Take pictures of important the people, places, and events at the wedding.
- Transfer the images from the camera to the computer. Depending on the camera and the software on the computer, this could mean taking the storage medium from the camera and inserting it into a reader, launching communication software, connecting the camera directly to the computer, placing the camera in a docking station, or selecting the transfer mode. Next, select a location to save to and file names for the images.
- If you choose to view, process, or organize your photos, you may have to use a separate software package for each task. Finally, you can print a copy of your images or share them with others via email or the creation of a web page.
- In the future, to send the wedding photos to a friend you would need to open an email application to type your message. Then you would have to launch another application to search for and view your images. If you find a relevant picture, you would then need to either upload it to your email account as an attachment or cut and paste them from the image viewer to the email editor [13].

1.3. Limitations and Assumptions

Unlike internet search engines, or even image retrieval engines for corporate use, consumers are not necessarily looking for a perfect match when they are using their photos. The process of browsing often inspires them and influences what they write about. For this reason, we are only using commonsense knowledge in a shallow way. ARIA searches for matches for a short period of time and then gives up gracefully, presenting what it found thus far. In terms of the design, it was more important that the system be responsive to users than that it finds the “optimal” reranking of images.

2. A User Scenario

2.1. A Wedding

When ARIA is started, the user is presented with a blank text editor on the left side of the window and a scrollable database of their images on the right-hand side. If no photos have been loaded into the program, the image frame will initially be empty. There are several activities that the user could choose to engage in from this initial state:

- Begin writing a new email message or web page
- Annotate images that are already in the system
- Load new images from his/her camera
- Add new knowledge to the personal repository via the repository interface.

Let us say that the user has recently taken pictures at the wedding of a friend, Meloni, which he/she wants to have added to ARIA for inclusion in a message. The user would remove the flash card from the cameras, insert it into the PCMLA reader and then insert that reader into the laptop. Immediately ARIA recognizes that there are new images and copies them into a central folder and adds them to the top of the scrollable image window. ARIA does not request any instruction regarding which images should be copied from the flash card or where they should be copied.

If the user wants to annotate images directly, he/she can double-click on the desired image and enter descriptive keywords. To enter useful personal information to assist in the use of these new wedding images, he/she could go up to the "Tools" menu and click the option for entering personal knowledge. Rather than do a general knowledge acquisition activity, he/she decides to use the "Enter a Fact" activity, which allows the user to simply enter any facts.

The user enters each of the following sentences in addition to any sentences that may already be present in the repository:

- Meloni is the bride.
- The groom is Mantell.

- The maid of honor is Angela.

After closing the activity interface, the user decides to begin writing an email to a friend about this particular wedding.

```
"Dear Sam,  
Last weekend I attended Meloni's wedding."
```

The agent continually extracts keywords -- words that may be meaningful -- by looking for the words in the neighborhood of the user's cursor. The default size of the neighborhood is set to three words, but this value can be altered in ARIA's options. After typing the above sentences, the user decides to include an image of Meloni, the bride at this wedding.

The initial query is [weekend, meloni, wedding]. Before immediately searching for images whose annotations match any of these keywords (as the original version of ARIA did), the agent attempts to expand the query to include keywords that are logically connected to those typed by the user. For example, the word "Meloni" should not only bring up images annotated with her name, but it should also trigger the connection that images labeled "bride" should be returned. Similarly, the keyword "wedding" should return pictures of the bride, groom, cake, etc.

In order to find these related keywords, both Open Mind, our repository of general knowledge, and our personal repository of knowledge are queried for each of the three keywords. If matches are found, keywords from the new sentences are added to the query. After this expansion is complete, the agent then continues on to search the photo database for matches.

The previous ARIA ranked images based on how many of the query words matched the image annotations. Those with the most matches were placed at the top of the list and those with the fewest matches were moved to the bottom of the list. ARIA with

commonsense handles the ranking process slightly differently. Because there are potentially many more words in the query string, this new version uses a weighting heuristic which will be discussed further in the methodology section.

This weighing algorithm takes into account whether a word was typed by the user or found through the query expansion process. The further away in the chain of expansion, the less weight is given to the word when re-ranking the images.

After the images have been reordered, the user then drags the top photo into the message after the first sentence and the original query keywords are added to the annotations for that image. In this case, "Meloni" is already annotated to the image, so only "weekend" and "wedding" are added. This will help with future retrievals and it also semi-automates the process of annotating photographs.

Next the user types another sentence:

```
"Meloni is a friend of mine from college."
```

Thinking that this is a piece of information that could prove helpful to the agent, the user highlights the sentence, right-clicks on it, and selects the menu option "Add to Repository". This is a quick way of adding specific facts deemed important by the user.

Finally, after completing the message, the user clicks the "Send Message" button and deciding to complete the session with ARIA, closes the application. Though every interaction with this software will not include all of the above steps, these options are available at any time during the session.

2.2. Greeting Cards

Another scenario in which ARIA with commonsense proved useful was for Hallmark. The user is presented with images of Valentine's Day cards, some with annotations, some without. As the user types an email message to a friend, the

system recommends cards related to what he/she is writing about. As seen below, when the user types the word “boyfriend” ARIA finds a photo of a card annotated with the word “love”.

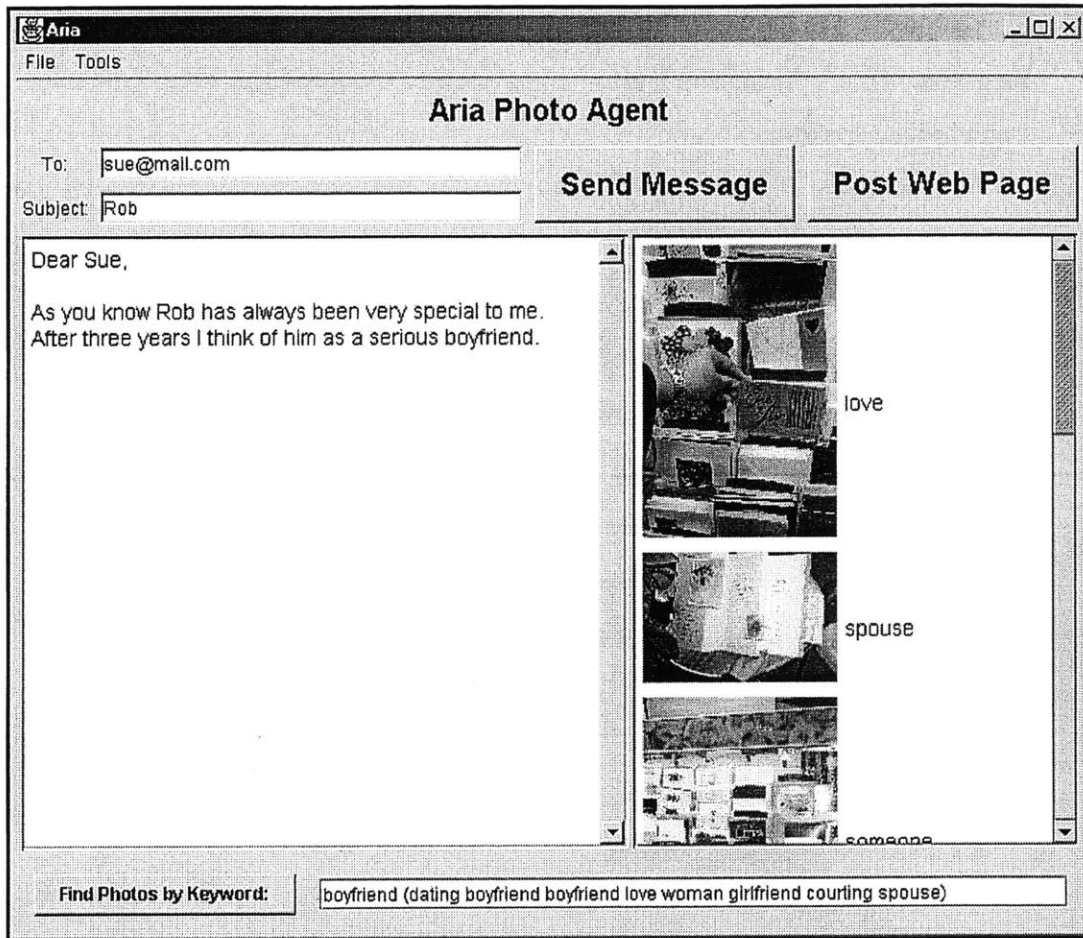


Figure 2.1: Using ARIA to locate greeting cards

During the first round of query expansion, we find the sentence

“A boyfriend is someone you love”

In this instance, ARIA can be seen as a product recommendation tool.

3. Methodology

3.1. Knowledge Repositories

3.1.1. Open Mind Commonsense

With direct keyword matching, many opportunities for the agent to make recommendations are missed. For example, it would be helpful if the agent knew

that when I am writing about my pet that it should bring up photos with annotations such as “dog” and “cat”. One would hope an agent would recognize the connection between these basic concepts, but in reality vast quantities of information are necessary in order to understand even the simplest of sentences. To bridge the gap between the query and the image annotations, more detail must be given. That information has to be provided to the agent via some type of knowledge repository.

For this project, general knowledge about the world comes from the databases of the Open Mind Commonsense Repository [19]. Open Mind allows for the elicitation of commonsense knowledge from non-expert internet users by prompting them, asking questions, and presenting them with lines of reasoning that they can confirm or repair. Open Mind approaches knowledge acquisition as a large-scale collaboration rather than as a carefully engineered task. Other famous commonsense systems such as CYC [10] and WordNet [Aslandogan, Miller] require having a team of knowledge engineers handcraft each piece of information.

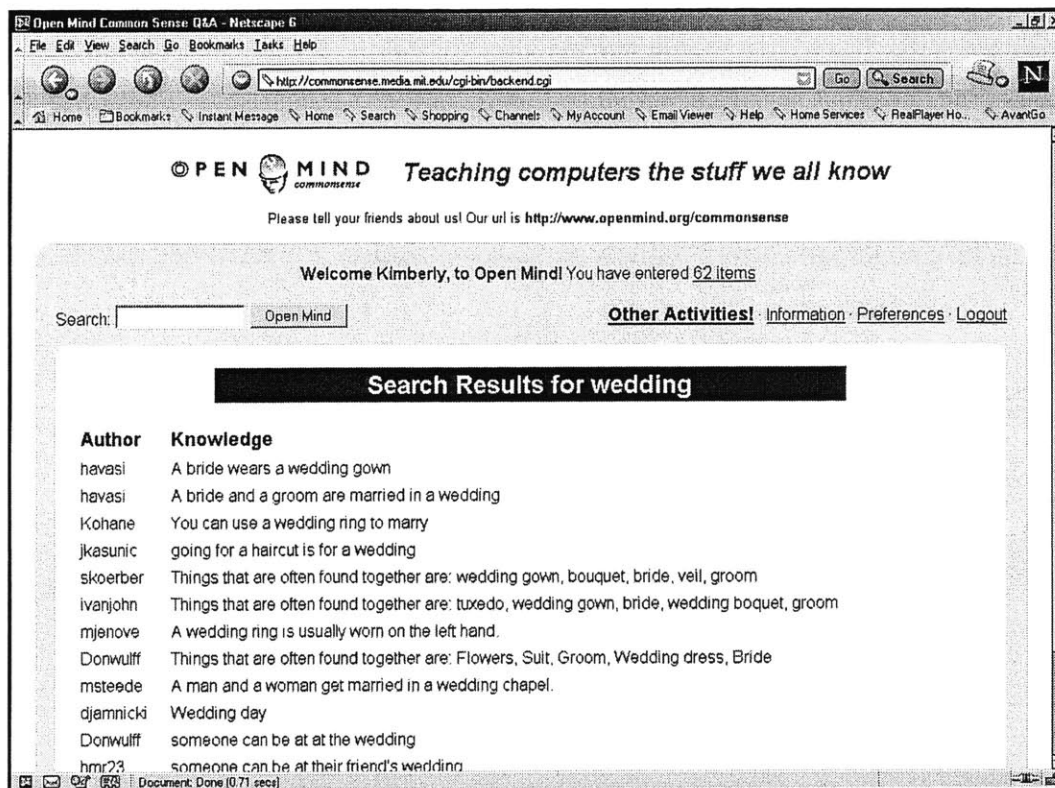


Figure 3.1: Open Mind web interface

By presenting the user with almost thirty different activities via a web interface (available at <http://www.openmind.org/commonsense/>), Open Mind potentially has access to the minds of millions. These activities collect many types of knowledge, including the following:

- **Spatial:** Describe a simple movie of a special event.
- **Relation:** Explain a relation between a pair of words.
- **Images:** Describe a photo of a typical scene.
- **Stories:** Supply a story given a story title.
- **Verbs:** Supply typical arguments of a verb.
- **Objects:** List objects that tend to appear together.
- **Explanation:** Explain an aspect of a short story.
- **Teleology:** Provide the function of an object.
- **Causality:** Give the likely effects of an action. [19]

At over 400,000 pieces of knowledge, Open Mind continues to grow daily. Each activity has a special interface for knowledge elicitation. Users are encouraged to enter simple English sentences that even a child could understand, thus their complexity is relatively low. Though Open Mind's data is less clean than that of its counterparts, using shallow parsing techniques, it is relatively easy to utilize for our purposes.

Examples of entries in the general knowledge base include:

- Something you find at a friend's wedding is wedding cake
- A bride and a groom are married in a wedding
- A steak house is a restaurant
- Something you find in a restaurant is a waiter
- An automobile is a car

We downloaded Open Mind as a plain text file from their website. Each sentence was read in, parsed, and stored in a table for easy access. After the initial loading of Open Mind, the table is saved to ARIA's program folder so that will be available to the next time the program is run. Each keyword has its own entry in the table. As a new sentence is added, it is first parsed in order to determine what keywords it contains. For each keyword, ARIA determines whether there is already an entry for that word. If so, the sentence is added to the corresponding vector of sentences that contained the keyword. If not, a new entry is created for that keyword and the sentence is then added to the entry's list of sentences.

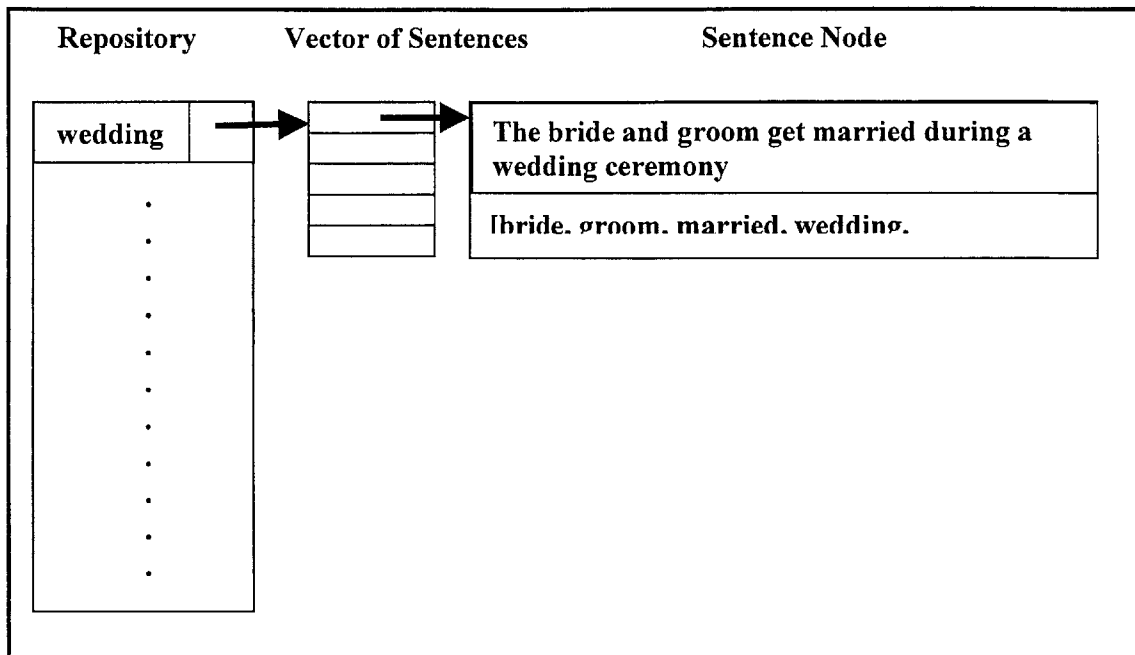


Figure 3.2: Repository table structure

The table structure is adequate for our needs, however, because Open Mind is so large, reading the whole table structure monopolizes the memory resources available on the system. A possibility for future development would be to store the contents of Open Mind Commonsense in a bona fide database. The most appealing solution would be to connect ARIA directly to the live version of Open Mind. This would free up the user's system and would take advantage of Open Mind's constant growth. Unfortunately, due to Open Mind's username and password structure and the lack of an API, this solution fell outside the time constraints of this stage of ARIA's development.

3.1.2. Personal Knowledge Repository

The repository of personal knowledge is structured similarly to Open Mind, but it is kept separate from the general knowledge that is applicable to all users. The personal knowledge base contains information about people, places, and events with which the user is familiar. This commonsense knowledge about the user is

useful for bridging the gap between general commonsense and the user's specific situation.

Examples of entries in the personal repository for our scenario include:

- The bride is Meloni
- The maid of honor is Angela
- The groom is Mantell
- Anthony is Mantell's son
- The ring bearer is Anthony

An interesting design issue for this project was how to obtain personal knowledge from the user. Unlike Open Mind, we do not have thousands of people available to enter information about each user. Also, we did not want to make the mistake of having a personified agent that dances around and distracts the user from the task he/she is trying to do. Combining general knowledge about the world and personal knowledge about the user can only occur if personal knowledge is available to the system. Even the smallest quantity can reap great results. But in the absence of specific knowledge, the general knowledge can still be very useful.

There are several approaches that have been incorporated in this version of ARIA in order to capture personal information. The first addition to the interface was a popup window that presents several activities (similar to those of Open Mind) to the user to encourage them to enter information such as "Describe a Picture" below.

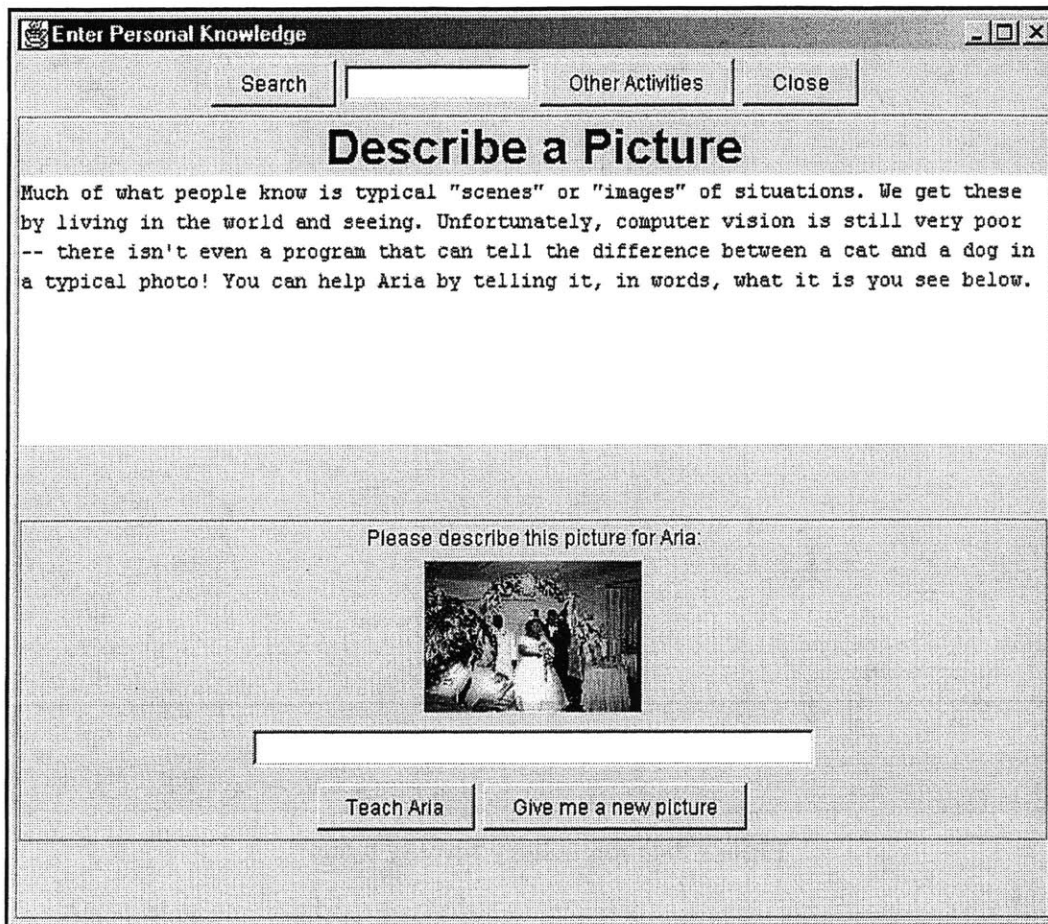


Figure 3.3: ARIA's "Describe a Picture" personal repository activity

These activities are a subset of those provided by Open Mind Commonsense's website. We also allow users to search the personal repository to examine what information it may contain about a particular topic. The idea behind this separate interface for eliciting personal knowledge is that when the user is not actively working on a message, he/she could go to the collection interface and complete activities that would increase the knowledge available for future query expansions.

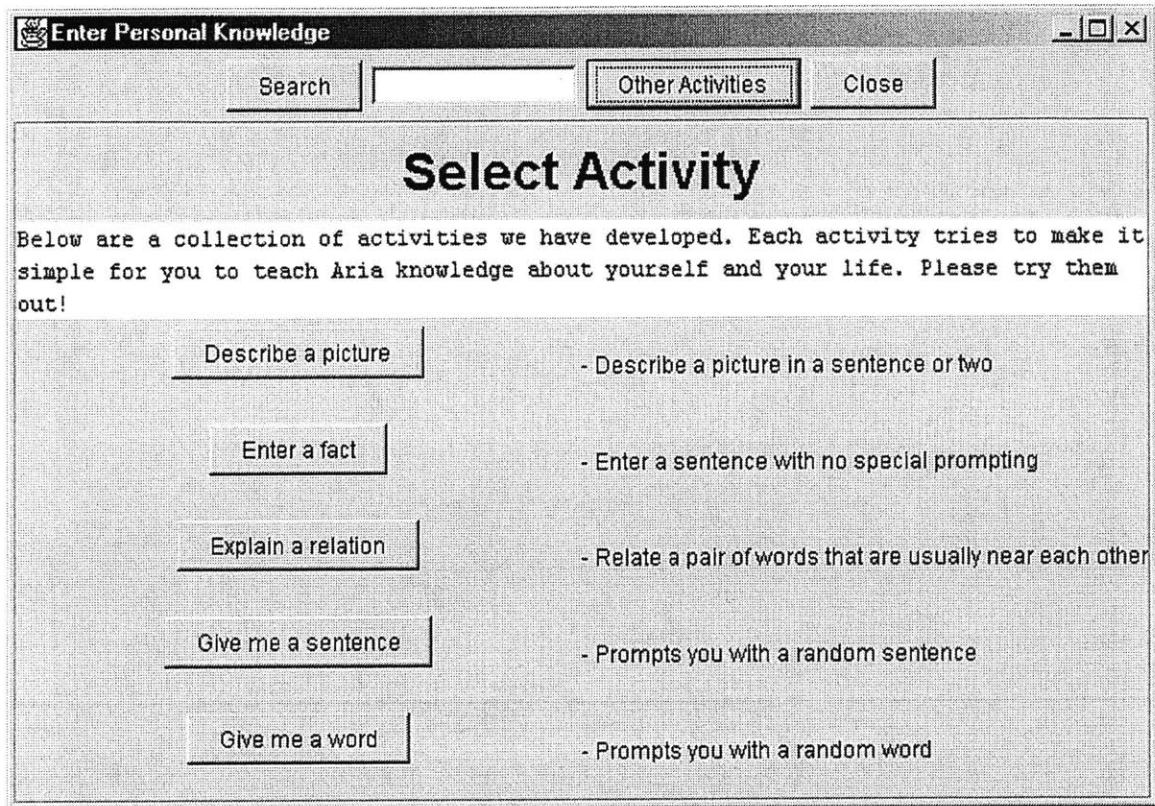


Figure 3.4: Repository activity menu

These activities are described below:

- **Describe a picture** randomly chooses one photo from the user's database, presents it to him/her, and asks for a single sentence description.
- **Enter a fact** presents an empty text box allowing users to enter any sentence with no prompting.
- **Explain a relation** randomly presents a pair of words usually found near each other and asks the user to explain how they are related to one another.
- **Give me a sentence** selects a sentence from the contents of the repository, prompting the user to enter a fact related to the sentence.
- **Give me a word** picks one of the keywords from the repository and allows users to share what they know about that topic.

Realistically speaking however, the user may choose not to enter information in this separate interface very often. These activities remove the user the message creation task for which they are using ARIA. These activities are also rather random in how they elicit information. For example, if the user decides to do the aforementioned "Describe a Picture" activity (Figure 3.3), he/she would be presented with an arbitrary image from the collection and asked to describe the contents of the photo. After that sentence is entered into the system, the user is presented with another random picture. This approach does not take into account the pictures that the user most recently used, or the text of the message he/she is currently working on. How can knowledge be obtained without interrupting the user's task-driven goals and without offering irrelevant options?

Our first solution to this issue was to allow users to add information to the repository as they create a message. While the user types, if he/she decides that a sentence or group of sentences from their message provide useful information for future use, he/she can highlight the text, right click on it (which brings up a small menu), and select the option "Add to Personal Repository". ARIA also allows users to employ the same select-and-click method to "Add to Open Mind", which will insert the text into their local copy of Open Mind's databases.

Another less disruptive solution was to add a "Tell me More" button. When clicked, the system displays a window that prompts the user to tell ARIA more about a specific topic. The topic inquired about is chosen using shallow query expansion on the keywords in the area surrounding the user's prompt. If the user types a sentence about having attended a wedding, clicking the "Tell me More" button might ask the user to say more about the bride, groom, or reception. This occurs because Open Mind has sentences connecting the word "wedding" to words like "bride, groom, flower girl, ceremony, reception, and bridesmaid".

3.2. Query Expansion

3.2.1. Bag-of-Words

ARIA represents the content of each sentence (whether in a repository or typed by the user) as the set of all words in that sentence. The words are viewed as having no structure or order. Information is extracted from the sentence simply by tokenizing words. Others are looking at how using natural language tools can yield benefits in the retrieval process [12]. Those tools allow the computer to use the sentence's structure as another source of information.

This basic "bag-of-words" approach [7] is then supplemented by the use of a "stop list". This list is used to remove common words such as "the", "and", and "he" which offer little semantic information. Finally, the bag of words taken from the user's text can be augmented by correlating bags of words found in our repositories and then compared against the user's image database for matches.

Let us revisit the wedding example to see how using commonsense knowledge to expand our queries can improve the search results. You attended a wedding and would like to send an email message to a friend containing the pictures you took. If you wanted to find an image of Angela, the maid of honor at this particular wedding, typing the following sentence in ARIA's text editor would result in no images of her being discovered:

```
"Angela, the bride's best friend, helped with
decorations."
```

There is a picture of Angela in the photo database, but it would not be found because the image is not annotated with the keyword "Angela", only the keywords "bridesmaids" and "flower girl".

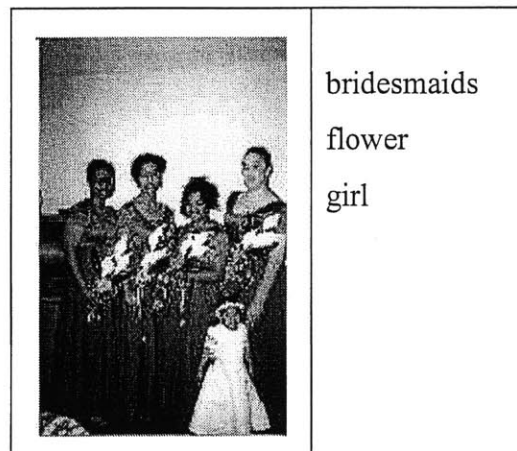


Figure 3.5: Picture containing Angela from database

Integrating personal information and commonsense knowledge would make it possible to locate the one image in the database containing Angela, the maid of honor. As the user types the sentence in the text-editing pane, the basic ARIA agent would search the database for the first keyword, “Angela”, and find nothing. If we assume the user has entered the sentence “The maid of honor is Angela” into the personal repository, then ARIA will be able to combine this data with general knowledge about weddings to find a picture of Angela.

First, ARIA searches both the personal repository and Open Mind for the keyword “Angela”. Nothing is found in Open Mind, but we do find the sentence “The maid of honor is Angela” in the personal repository. The program would then extract all possible keywords from the sentence (“maid” and “honor” in this case) and both databases would again be queried for sentences containing the new keywords. In searching Open Mind, ARIA retrieves the fact “The maid of honor is one of the bridesmaids.” Using associative query expansion loosely based on a basic taxonomic inference rule (Figure 3.6) ARIA can then combine this fact with the user’s personal information to find the desired keyword (Figure 3.7).

X is a Y Y is a Z → X is a Z

Figure 3.6: Basic query expansion rule

Angela is a maid of honor The maid of honor is a bridesmaid → Angela is a bridesmaid
--

Figure 3.7: Query expansion rule applied to the example

The new fact is stripped for keywords and the ARIA database is then searched using the keyword “bridesmaids”. The image containing Angela is found and placed at the top of the list of photos.

Unfortunately, since we are not using strict logical inference, it is possible for the system to make a mistake. Two pieces of information can be combined to produce an illogical conclusion. An example of this can be seen in Figure 3.8. Though both statements are true under certain circumstance, the conclusion reached is inconsistent.

An inexpensive wedding gown is a rare thing. A rare thing is expensive. → An inexpensive wedding gown is expensive.

Figure 3.8: Possible incorrect conclusion using query expansion

3.2.2. Weighting Heuristic

A necessary addition to the query expansion process is that the algorithm assigns a value to each of the words placed in the query string. Words found after several steps of expansion should not have the same weight as words typed by the user

because they are possibly less relevant. If there is a direct match for a word typed by the user, that photo should be ranked higher than a picture that happens to contain more of the inferred keywords.

To ensure this in fact occurs, we added a weighting heuristic such that words are given a specific value (words typed by the user have the highest weight, words found in the first round of inference will be ranked lower, decreasing down to the last round of inference). When ARIA reranks the photos, those with the highest combined weight for matching words are moved to the top of the list.

First, the weight of each keyword is calculated with the formula below. The base value of 0.3 was chosen such that a photo containing only one of the initial keywords would be able to outrank a photo containing 2 or 3 secondary keywords. This processes of assigning weights is done as new keywords are added to the query during each expansion step.

$$\text{keyword_weight} = 0.3^{\text{expansion_level}}$$

After the agent has completed the expansion phase, the initial value for each photo is set to zero. Then each photo in the user's database is compared against the keywords found in the course of expansion. If there is a match, the photo's value is updated as follows:

$$\begin{aligned} \text{old_value} &= \text{value} \\ \text{value} &= \text{old_value} + \text{keyword_weight} \end{aligned}$$

Finally, all of the images are reranked based on the combined value of the matching keywords' weights.

For example, given the pictures in Figure 3.9, how would the photos be reordered when the sentence,

"Last weekend I saw my friend Meloni,"

is typed? Using a stop list, the word "Meloni" is selected as one of the keywords found in the sentence. Pictures 1, 2, and 4 all show Meloni, the bride, but only the first one is annotated with her name. Using query expansion with the weighting heuristic, we get the following results.





Photo #	Photo	Annotations
1		meloni procession
2		bride parents
3		flower girl
4		bride groom dance

Figure 3.9: Sample image database

During the first round of searching for the word “Meloni” in both Open Mind and in the personal repository, we find the sentence

“The bride is Meloni.”

which gives us the new keyword “bride”. Searching Open Mind and the personal repository with the new keyword results in the following sentences being found:

Open Mind

“The maid of honor is the sister or best friend of the bride.”

“The wedding guests are friends and family of the bride and groom.”

Personal Repository

“The bride is Meloni.”

“The bride’s mother is Bertha.”

“The bride’s father is Norman.”

Each sentence is then passed through the keyword extraction algorithm, which takes all words that are not on the stop list and places them into an array of keywords. Based on the level of expansion during which each keyword was found and the equation for calculating keyword weights, the values of each of the keywords in our expanded query are below:

Expansion Level	Keyword	Value
0	meloni	1
1	bride	0.3
2	maid	0.09

2	honor	0.09
2	matron	0.09
2	sister	0.09
2	wedding	0.09
2	guests	0.09
2	family	0.09
2	groom	0.09
2	bertha	0.09
2	norman	0.09

Figure 3.10: Values for each keyword in the expanded query

These values cause the images to be reranked from the closest match to the query to the furthest match based on the combined weights for matching annotations.

New Photo Order	Combined Value of Matches
Photo 1	$1 + 0.3 = 1.3$
Photo 4	$0.3 + 0.09 = 0.39$
Photo 2	0.3
Photo 3	0.0

Figure 3.11: New ranking of photos based on combined keyword weights

3.3. Interface Changes

When interacting with software agents, it is vital that the user trust what the agent does on his/her behalf [14]. One way to encourage that trust is to make the actions of the agent transparent to the user. By this, I mean that the user understands the general decision making process of the agent without necessarily knowing all the inner workings of the program.

The first version of ARIA was fairly straightforward for users to understand. As the user types in the text-editing pane, the agent is constantly observing, his/her actions awaiting possible opportunities to recommend images. ARIA would compare words in the neighborhood of where the user was typing with the words annotated to the images in the user's photo database. If an exact match was found, the images were re-sorted such that the closest matches would be visible at the top of the photo pane. Though users did not know all the details of how the program worked nor the algorithms used, they understood the direct correlation between what they typed and the images that were found by the agent.

In an attempt to maintain this level of transparency, there are a couple of new features that have been added to this version. The first is a display of the reasoning steps taken by the new query expansion module (Figure 3.12). Though users will probably not care to see this trace every time they use the program, it will help in building the level of trust with the agent and in understanding what led the agent to re-rank the photos in the manner it did. This query trace is an integrated part of ARIA's interface, but there is an option that allows users to either display hide the frame. The default is to hide the trace window.

For our example from Section 2

"Dear Sam,

Last weekend I attended Meloni's wedding."

after typing the word "Meloni", the resulting query trace is would be

```

-----
<<<<<1st ROUND>>>>>
Original Query: [attended, meloni]
word 0: attended
word 1: meloni
      PERSONAL:
      The bride is Meloni
Keys1: [bride]
<<<<<<2nd ROUND>>>>>
word 0: bride
      OPENMIND:
      The maid of honor or matron of honor is the sister

```

```

    or best friend of the bride
OPENMIND:
    The wedding guests are friends and family of the
    bride and groom
PERSONAL:
    The bride is Meloni
PERSONAL:
    The bride's mother is Bertha
PERSONAL:
    The bride's father is Norman
Keys2: [maid, honor, matron, sister, wedding, guests,
    family, groom, bertha, father, norman]

All new keys: [bride, maid, honor, matron, sister, wedding,
    guests, family, groom, bertha, father, norman]
-----

```

Round one consists of taking and querying both Open Mind and the user's personal repository. If there are no matches, ARIA moves on to the next word in the original query. If the word is found, the trace labels which repository it came from. A maximum of three sentences from each repository can be selected. At the end of round one, each sentence is stripped of its keywords, which are then placed on the array Keys1.

Round two of the query expansion process involves querying the repositories for the new keywords. Any keywords discovered in this step are then placed in the array Keys2. The arrays, Keys1 and Keys2, are combined with the original query terms to create the expanded query. The new query string is then compared against the image annotations such that closer matches are ranked higher in the photo window.

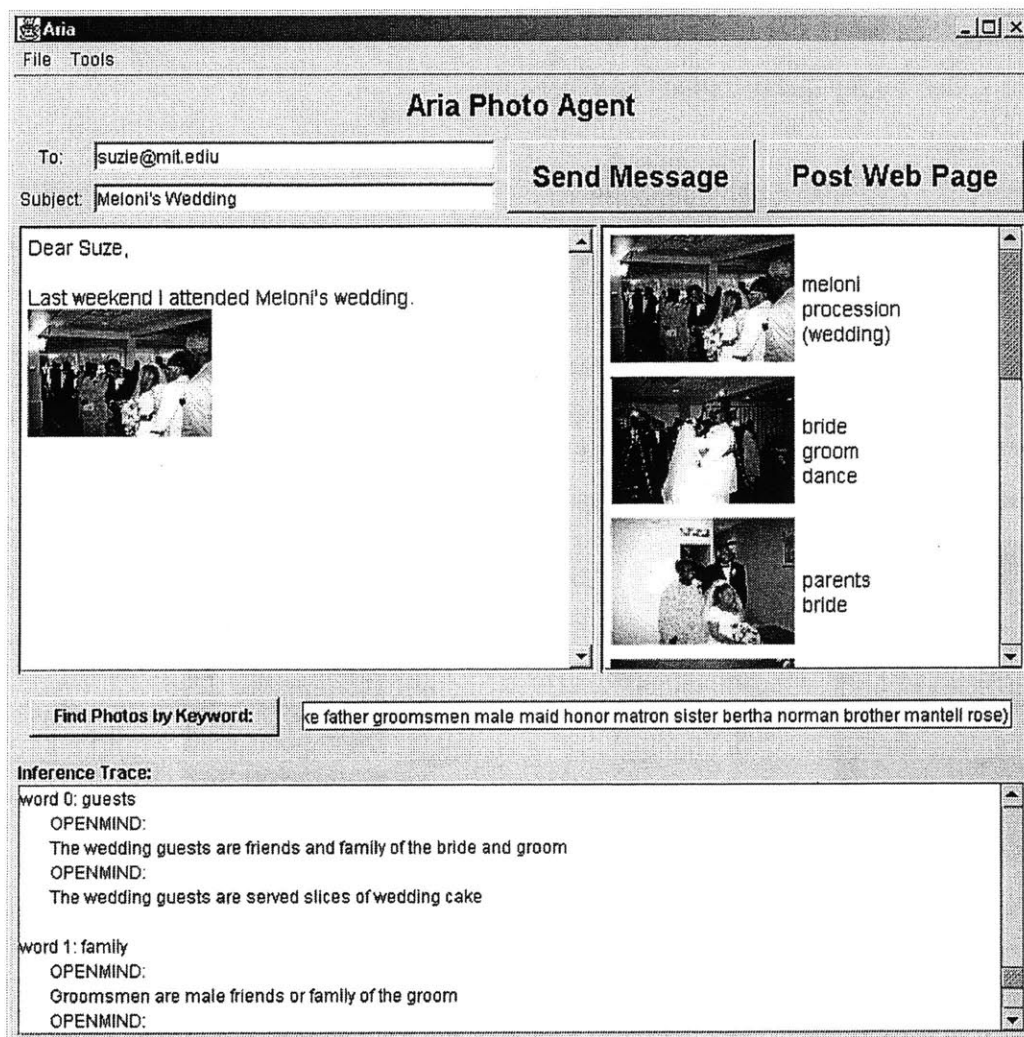


Figure 3.12: ARIA with trace window displayed

Another useful tool in building trust is to make it obvious what changes in the interface reflect the user's actions and what actions the agent has taken. Previously users have complained that it was difficult to notice when a keyword had been automatically annotated, and thus possible incorrect annotations frustrated them because they felt it was difficult to rectify the situation.

To remedy this, parentheses are used to distinguish between information provided by the user and information produced by the agent (Figure 3.13). This low-tech solution places parentheses around words annotated to an image as a result of an image being dragged into the user's text. Parentheses are also used in the query text area (at the

bottom of the interface) to distinguish between query words typed by the user and those added by the query expansion algorithm.

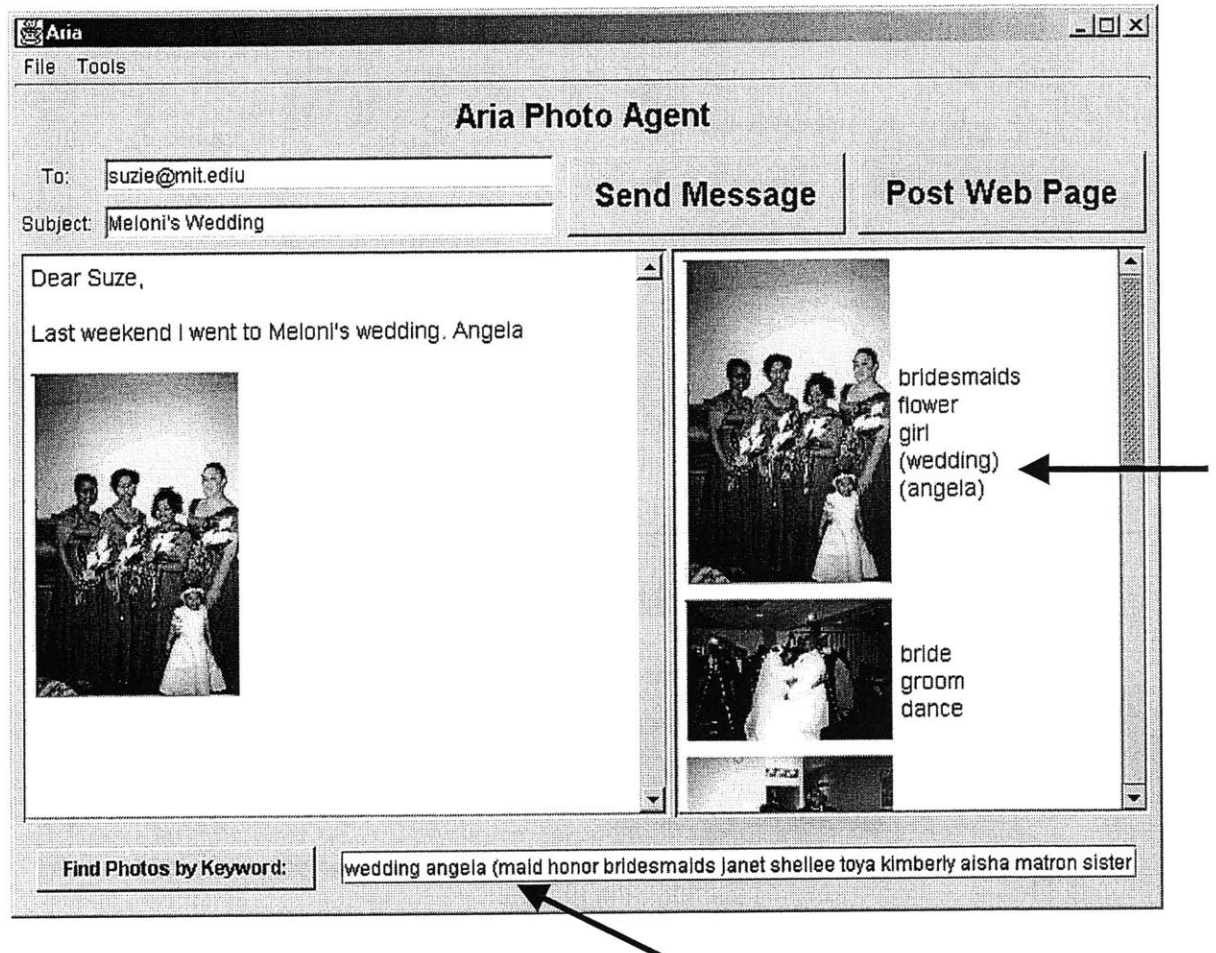


Figure 3.13: New interface showing the use of parenthesis

4. Related Work

4.1. Digital Image Retrieval

In terms of related work in the area of digital image retrieval and annotation three systems stand out: PhotoFinder, FotoFile, and Watson. They all seek to assist non-expert users in finding information, but they attack the problem of retrieval from different directions. PhotoFinder focuses on making the annotation process simpler for users. FotoFile allows for automatic image annotations and browsing. Watson takes into account the

user's context and goals in determining what results should be presented to the user.

4.1.1. PhotoFinder

Because annotating photographs can be such a tedious process, PhotoFinder [18] aims to make the task easier and less prone to user error. Without accurate descriptions of an image's content (people, place, date, time, etc.), locating pictures can be an arduous task. Also, having misspellings or variations of names can negatively affect search results. While ARIA tries to remedy this issue by incorporating knowledge that Bill, Billy, and William are all the same person, PhotoFinder standardizes annotations by keeping users from having to type names repeatedly for multiple images.

Photographs are organized into a library consisting of collections of individual images. Photos can participate in multiple collections. The collections as well as the individual pictures can be annotated using date, location, and free text fields, which are stored in a database. The interface is split into three panels:

- **Library viewer:** displays one representative image for each collection on top of a stack representing the quantity of images in that collection
- **Collection viewer:** displays thumbnail images of the photos in a collection
- **Photo viewer:** displays a single photo from a collection

PhotoFinder employs direct annotations – labels that can be dragged directly into a photo. Users are presented with a scrollable list of names from which to choose. Once a name has been entered via the keyboard, it never has to be entered again. If an identity label has been inserted into a picture, its presence is noted by the database and it will be used for retrievals. These name labels can either be viewed or hidden simply by clicking a Show/Hide checkbox. To see all photos that include a specific person, the user can drag their name from the list into the

Collection viewer and PhotoFinder locates and displays any photo containing that name label.

4.1.2. FotoFile

FotoFile [9] is an application designed for multimedia organization and retrieval. As with PhotoFinder, images and movie clips are organized into photo albums, each having a representative cover image to allow for easy visual identification. Each photograph may be annotated with information that is meaningful to the user. The predefined metadata attributes include

- Creation date
- Location
- Subject
- People
- Title
- Description

Users can perform bulk annotations of images. They can select multiple attributes and multiple images, then click the Annotate button. This will associate all of the selected attributes to each of the desired images. This feature streamlines the process of manually annotating photographs.

This system also utilizes face recognition to partially automate annotation. When FotoFile comes across an image containing the faces of people it does not recognize, the user can enter the identity of each person. Any time that face is seen in subsequent photos, the identity will be associated with the new image. User validation is required before this association occurs so that false positive and negative results are avoided. He/she simply confirms the selected identity or corrects the system's choice.

FotoFile also focuses on browsing and visualizing data. While information retrieval professionals tend to perform very targeted searches, consumers both expect and enjoy the process of scanning through their photos. In addition to allowing users to “flip” through the pages of their albums, this program also provides a hyperbolic tree to cluster images based on the similarity of their annotations as well as their visual characteristics. This feature enables users to view how their images cluster together.

4.1.3. Watson

Watson [2], on the other hand uses contextual information collected from the document currently being edited by the user to retrieve related documents and photos. In order to compute the relevance of data returned by the distributed information repositories, a weighting algorithm is utilized. The seven heuristics for this algorithm are

- 1) Remove all stop words
- 2) Value words found frequently in the user’s document
- 3) Value words that are stylistically emphasized
- 4) Value words at the beginning of the document more than those at the end of the document
- 5) Devalue words that are stylistically de-emphasized
- 6) Value the words in lists equally (an exemption from heuristic 4)
- 7) Ignore words that do not reflect the document’s subject

The main focus of Watson is text retrieval in which the document’s content can be garnered from its text. Application adaptors gain access to an application’s representation of data, which is represented as normal, emphasized, de-emphasized, or list words. Then using the above heuristics, the document is converted into a query. Watson also employs information adaptors that perform searches using the query and collect the results.

Though Watson can use text entered as a caption to query an image search engine and allows users to drag and drop those photos into their document, the program does not in turn use that same context to annotate the pictures. This causes the program to miss an opportunity to assist future searches with better annotations. With images, the application is stuck with the original annotations available for each image.

4.2. User Interface Agents

ARIA follows strongly in the tradition of user interface software agents. These agents are likened to having a personal assistant that looks at the user's habits and aims to assist him/her in achieving his/her goals. The agents most related to ARIA are the Remembrance Agent and Letizia. Both observe the user and draw on that information to perform tasks on the user's behalf.

4.2.1. Remembrance Agent

Similar to Watson, the Remembrance Agent [17] continually looks over the shoulder of the user, watching for opportunities to present information that may be of use in the current situation. The agent observes what the user reads and types and looks for old emails, personal files, and on-line documents which are relevant to the user's context. The information is then presented unobtrusively in the interface so as not to distract the user from his/her primary task.

Suggestions are presented as one line summaries at the bottom of the user's screen and are updated about every ten seconds or so as the user's context changes. The aim of the Remembrance Agent is not to present "the" most relevant document, but to display a range of possibilities. These possibilities can augment human memory by reminding the user of important information associated with what they are currently doing. ARIA's method of information retrieval is built on the Remembrance Agent's platform.

4.2.2. Letizia

Letizia [11] is an interface agent that makes real-time suggestions for web pages that may be of interest based on the page currently being viewed. The program performs a breadth-first search of the links on the current page in order to find other related pages that may appeal to the user. Again, the agent is not focused on locating the “optimal suggestion”. The goal is to make the best use of limited query time and the user’s time and attention.

Rather than acting on the user’s behalf, Letizia simply presents suggestions in a low cost manner such that if he/she desires to explore those proposed ideas he/she can. The program turns makes web browsing into a collaborative effort and keeps the user from having to explore every link on a page or its connecting pages to discover something interesting.

4.3. Knowledge Repositories

4.3.1. CYC

CYC [10] is one of the best-known knowledge repository projects having begun in 1984 and continuing today. Consisting of more than a million handcrafted axioms and millions more having been inferred and cached into its knowledge base, it is extremely large. In order for computers to have the ability to reason on the level of even a child, it has always been accepted that large repositories of information would need to be created. CYC attempts to organize information in such a way that logical inference can be applied to it.

Written in the language CYCL, the designers entered about 10^6 assertions using a vocabulary of around 10^5 atomic terms. These axioms attempt to cover

- Causality
- Time
- Space
- Substances
- Intention

- Contradiction
- Uncertainty
- Belief
- Emotions
- Planning

Each assertion exists within a context distinguished by the assumptions that are made. Assertions can be imported from one context to another, but it is necessary to specify which assumptions differ from the original context. CYC places every assertion into at least one explicit context.

4.3.2. WordNet

WordNet [15], an online lexical database, organizes words into sets of synonyms linked via their semantic relationships. Pointers connect different word forms providing the necessary links. The semantic relationships supported are

- **Synonyms:** symmetric relations between words
- **Antonyms:** opposing relations between words
- **Hyponyms:** hierarchical relations between words (sub relations)
- **Meronyms:** relations between a whole and its parts
- **Troponyms:** manner (same for verbs as hyponyms are for nouns)
- **Entailment:** relations between verbs

WordNet is not a database of commonsense knowledge per se, but it is a useful tool in for natural language understanding. It provides an electronic dictionary: nouns, verbs, adverbs, and adjectives and the relationships that establish the meaning of words. Each set of synonyms represents a lexical concept.

4.4. Knowledge-Based Applications

4.4.1. SensiCal

SensiCal [16] is a commonsense calendar application. It aims to assist the user by automatically filling in missing information related to an event, and it produces

error messages warning that the user has attempted to create an event that is not logical (such as a lunch date at 4:00 in the morning or dinner with a vegetarian at a steak house). The program runs as either an extension to ical or as a batch process on exported calendar data.

Every time a new item is added to the calendar the following three steps are performed:

- 1) Find any pertinent information in the text entered by the user. This is accomplished by parsing dates, times, names, and phone numbers.
- 2) Fill in omitted information by providing intelligent defaults. SensiCal fills in the duration of events automatically using the typical length of a specific activity.
- 3) Indicate potential problems to the user. These includes violations of commonsense regarding
 - Space and time (a person can not be in two places at the same time)
 - Venues (you can not eat at an establishment that is closed)
 - Activities (lunch is eaten for about an hour beginning around noon)
 - Settings (people purchase items at stores that sell those items)
 - Food (avoid restaurants that mostly serve food you do not eat)
 - Social interactions (people are likely to attend events thrown by their friends and family)

The knowledge base for this application comes from ThoughtTreasure, developed alongside SensiCal. Representing data both declaratively and procedurally, it contains pieces of knowledge such as

- Soda is a drink.
- Dinner lasts about 2 hours.
- The evening extends from about 5pm to 9pm.

This repository does not attempt to represent all commonsense knowledge – only to prevent obvious scheduling mistakes, point out issues, and provide defaults that make sense.

4.4.2. SCORE

SCORE [1] provides a visual query tool and a search engine. The query interface, CANVAS, contains an icon palette and a query window. The icons represent objects, each containing their own set of attributes whose values can be defined by the user or selected from a list. This tool was designed to help non-expert users intuitively query the system.

The images are stored in a database with their corresponding meta-data annotations. SCORE uses WordNet to expand both the user’s queries and the annotations from the database. A distance metric is used to gauge how close of match there is between the user’s query and the images in the database. Exact matches yield a higher similarity value than matches that have more distant relations.

4.4.3. Commonsense Reasoning ARIA with Adaptive Linking

Developing concurrently with my research was an effort to combine the commonsense knowledge of Open Mind with natural language parsing techniques to improve retrieval and annotation [12]. These techniques are used to extract roles/concepts from text such as “who, what, where, and when”. Concepts, unlike keywords, are not sensitive to morphological variations (such as the difference between singular and plural nouns) or abbreviations.

The keywords extracted from the user’s text are converted to concepts using a morphological tool and a synonym dictionary. Proper names are determined using a first name dictionary and geographical information is found in databases, both garnered via the web. Common places, things, and events are extracted from Open Mind’s database. Then heuristics are used to determine which concepts are

relevant to a particular image and should be added as annotations. Open Mind's commonsense relations are also used to expand the annotations during querying.

To obtain the necessary concepts, a set of 20 mapping rules are applied to each sentence in Open Mind producing 50,000 relations. They fall into the categories listed below:

- Classification
- Spatial
- Scene
- Purpose
- Causality
- Emotion

These relational structures are converted into a spreading activation network made up of 30,000 concepts and 160,000 directed edges. Two things determine the weight associated with each edge: reinforcement and popularity. Reinforcement occurs when mutually connected concepts form a cluster enhancing the relevance of each node. Popularity implies that the more children a node has, the less likely each one is to be relevant.

5. User Evaluation

During user testing by Kodak, ARIA was compared with Kodak's Picture Easy software to gauge its usefulness in organizing photos and sending emails. With ARIA, users appreciated having the images visible as they wrote their messages. They occasionally altered the direction of their message based on the images returned by the agent. This collaborative aspect of the interface intrigued users.

While they found the keyword matching of great use in locating photos used previously, one participant of the study ran into problems when she typed "artwork" and her image was annotated "painting". This directly speaks to the need for a

commonsense ARIA. By expanding queries to locate words that are synonyms or that are related, these missed opportunities are now less likely to occur.

6. Conclusion

As digital photography continues to grow and computers become more advanced and complex, intelligent software agents will be needed to reduce the complexity of user interfaces. By integrating the use, annotation, and retrieval of images, ARIA takes a step in the right direction by streamlining the process and hiding unnecessary details from the user.

With the addition of both commonsense knowledge and a personal knowledge repository, ARIA becomes an even more valuable tool for the recreational digital photographer. Using very shallow query expansion, many images that had previously been missed by the program are now locatable. An assistant who knows you is always preferable to one who is uninformed, especially in the case of software agents. Knowledge is power.

7. Acknowledgements

I would like to thank my advisor, Henry Lieberman, for his patience, guidance, and insight. I would also like to thank Push Singh and the MIT Media Lab's Commonsense Reading Group for their helpful comments and suggestions on the ARIA project. Thanks also to Kodak for the use of a DC265 zoom camera. This work was partially funded by the MIT Media Lab, EECS Department, and the Graduate Students Office.

References

1. Aslandogan, Y.A., Their, C., Yu, C.T., Zou, J., and Rishe, N.: Using Semantic Contents and WordNet in Image Retrieval, ACM Special Interest Group on Information Retrieval (SIGIR '97), (1997), pp. 286-295.
2. Budzik, J. and Hammond, K.J.: User Interactions with Everyday Applications as Context for Just-in-Time Information Access. ACM Conf. Intelligent User Interfaces (IUI 2000), ACM Press, New York, (Jan 2000), pp. 44-51.
3. Canto Software, [cited 2002 January 27], Available HTTP: <http://www.canto.com/>.
4. Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., and Malik, J.: Blobworld: A System for Region-Based Image Indexing and Retrieval, Third International Conference on Visual Information Systems, (June 1999).
5. Extensis Corporation, [cited 2002 January 27], Available HTTP: <http://www.extensis.com/>.
6. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Quian, H., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P.: Query by Image and Video Content: The QBIC System, IEEE Computer, 28(9), (September 1995), pp. 23-32.
7. Hirschman, L., Light, M., Breck, E., and Burger, J.D.: Deep Read: A Reading Comprehension System, Proceedings of the 37th annual meeting of the Association for Computational Linguistics (ACL), College Park, MD, (June 1999).
8. Imospace Systems Corporation, [cited 2002 January 27], Available HTTP: <http://imspace.com/>.
9. Kuchinsky, A., Pering, C., Creech, M.L., Freeze, D., Serra, B., and Guizdka, J.: FotoFile: A Consumer Multimedia Organization and Retrieval System. ACM Conference on Human-Computer Interface (CHI '99), 37(7), (May 1999), pp. 496-503.
10. Lenat, D.: CYC: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM, 38(11), (November 1995), pp. 33-38.
11. Lieberman, H.: Autonomous Interface Agents. ACM Conference on Human – Computer Interface (CHI 97), Atlanta, (March 1997), pp. 67-74.
12. Lieberman, H. and Liu, H.: Adaptive Linking between Text and Photos Using Common Sense Reasoning. Proceedings of the 2nd International Conference on

Adaptive Hypermedia and Adaptive Web Based Systems (AH2002), Malaga, Spain, (2002).

13. Lieberman, H., Rosenzweig, E., and Singh, P.: ARIA: An Agent for Integrated Annotation and Retrieval of Images. *IEEE Computer*. (July 2001), pp.57-61.
14. Maes, P.: Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7), (1994), pp.30-40.
15. Miller, G.A.: WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), (November 1995), pp.39-41.
16. Mueller, E.T.: A calendar with commonsense. *Proceeding of the 2000 International Conference on Intelligent User Interfaces*, ACM, (2000), pp. 198-201.
17. Rhodes, B. and Starner, T.: The Remembrance Agent: A Continuously Running Automated Information Retrieval System. *The Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)*, London, UK, (April 1996), pp. 487-495.
18. Shneiderman, B. and Kang, H.: Direct Annotation: A Drag-and-Drop Strategy for Labeling Photos. *The Proceedings of the International Conference on Information Visualization (IV 2000)*, London, England, (July 2000), pp. 88-95.
19. Singh, P. The Public Acquisition of Commonsense Knowledge. (2001). Available: <http://www.openmind.org/commonsense/pack.html>.

Appendix A: Personal Repository Contents

Below is a listing of the knowledge in the personal repository used for the examples. Each sentence was written on its own line within a plain text file. This file was read in and parsed the first time ARIA was run. The information could have been entered one fact at a time using the personal repository interface described previously. As can be seen, it does not take very much personal knowledge to reap great benefits in image retrieval.

The bride is Meloni
The groom is Mantell
The bride's mother is Bertha
The bride's father is Norman
The groom's mother is Rose
The groom's father is not present at the wedding
The best man is Jason
The maid of honor is Angela
The bridesmaids are Angela, Janet, Shellee, Toya, Kimberly, and Aisha
The groomsmen are Jason and Melvin
The flower girl is Desiree'
The ring bearer is Anthony
Anthony is Mantell's son
Meloni wore a white gown
Mantell wore a white tuxedo
Desiree' is Aisha's daughter

Appendix B: Subset of Open Mind Contents

This is a subset of the total Open Mind database that was used for testing purposes. As with the personal repository, each statement was typed on a separate line in a plain text file and imported into ARIA the first time it was run.

The bride and groom get married during a wedding ceremony
 The groom usually wears a tuxedo
 A flower girl is a child
 The ring bearer is a child
 The flower girl tosses flower petals on the ground for the bride to walk on
 The ring bearer carries the wedding rings on a pillow
 Groomsmen are male friends or family of the groom
 One of the groomsmen is called the best man
 The best man is usually the brother or best friend of the groom
 The best man stands next to the groom during the wedding ceremony
 One of the bridesmaids is called the maid of honor or the matron of honor
 The maid of honor or matron of honor is the sister or best friend of the bride
 The maid of honor or the matron of honor stands next to the bride during the wedding ceremony
 The wedding party consists of the bride, groom, bridesmaids, groomsmen, flower girl, and ring bearer
 The father of the bride walks her down the aisle during the procession
 The bridesmaids carry bouquets of flowers
 A reception follows the wedding ceremony
 The bride and groom feed each other cake at the reception
 The best man makes a toast to the bride and groom at the reception
 The bride and groom share the first dance at the reception
 The bride has a special dance with her father at the reception
 The groom has a special dance with his mother at the reception
 The groom throws the bride's garter at the reception
 The wedding guests are friends and family of the bride and groom
 The wedding guests are served slices of wedding cake
 Guests eat, drink, dance, and mingle at the reception
 Guests tap their glasses at the reception to get the bride and groom to kiss
 A buddy is a friend
 A pal is a friend
 A confidant is a friend
 A best friend is a friend
 A chum is a friend
 A woman's female friend is called a girlfriend
 A person you go out with is a date
 A date is an escort
 In dating, the woman is called the girlfriend

In dating, the man is called the boyfriend
Courting is dating
Wooing is dating
A boyfriend is a significant other
A girlfriend is a significant other
When engaged, the significant other is a fiancée
A man and woman date before getting engaged
A colleague is someone you work with
A coworker is a colleague
A supervisor is a colleague
A supervisor is a boss
An employee is a colleague
A subordinate is a colleague
A parent has a child
Children with the same parents are siblings
A female child is a daughter
An immediate family consists of parents and children
A grandparent is the parent of the parent of a child
A male grandparent is a grandfather
A female grandparent is a grandmother
An aunt is a sister of a parent
The maid of honor is one of the bridesmaids

Appendix C: Query Trace for User Scenario

What follows is the trace produced by the user scenario discussed previously. White space and punctuation is taken to represent the end of a word. Whenever the end of a word is signaled, ARIA performs query expansion and a reranking of the photos. During the first round, each word in the original query string is expanded using knowledge from both the personal and general repositories. New keywords are extracted and placed in the array Keys1. The second round consists of expanding each of the words in Keys1 and placing new keywords in Keys2. These words are combined into a single array and used to reorder the image database. Words in the original query have the highest value, then words found in the first round of expansion, and finally words found in round two.

```
-----
<<<<<1st ROUND>>>>
Original Query: [dear]
word 0: dear
Keys1: []
<<<<<<2nd ROUND>>>>>
-----
```

```
-----
<<<<<1st ROUND>>>>
Original Query: [dear, sam]
word 0: dear
word 1: sam
Keys1: []
<<<<<<2nd ROUND>>>>>
-----
```

```
-----
<<<<<1st ROUND>>>>
Original Query: [dear, sam]
word 0: dear
word 1: sam
Keys1: []
<<<<<<2nd ROUND>>>>>
-----
```

```
-----
<<<<<1st ROUND>>>>
Original Query: [dear, sam]
word 0: dear
word 1: sam
Keys1: []
<<<<<<2nd ROUND>>>>>
-----
```

```

-----
<<<<<1st ROUND>>>>
Original Query: [dear, sam]
word 0: dear
word 1: sam
Keys1: []
<<<<<<2nd ROUND>>>>>
-----

```

```

-----
<<<<<1st ROUND>>>>
Original Query: [dear, sam]
word 0: dear
word 1: sam
Keys1: []
<<<<<<2nd ROUND>>>>>
-----

```

```

-----
<<<<<1st ROUND>>>>
Original Query: [sam, weekend]
word 0: sam
word 1: weekend
Keys1: []
<<<<<<2nd ROUND>>>>>
-----

```

```

-----
<<<<<1st ROUND>>>>
Original Query: [weekend]
word 0: weekend
Keys1: []
<<<<<<2nd ROUND>>>>>
-----

```

```

-----
<<<<<1st ROUND>>>>
Original Query: [weekend, attended]
word 0: weekend
word 1: attended
Keys1: []
<<<<<<2nd ROUND>>>>>
-----

```

```

-----
<<<<<1st ROUND>>>>

```

Original Query: [attended, meloni]

word 0: attended

word 1: meloni

PERSONAL:

The bride is Meloni

Keys1: [bride]

<<<<<<2nd ROUND>>>>>>

word 0: bride

OPENMIND:

The maid of honor or matron of honor is the sister or best friend of the bride

OPENMIND:

The wedding guests are friends and family of the bride and groom

PERSONAL:

The bride is Meloni

PERSONAL:

The bride's mother is Bertha

PERSONAL:

The bride's father is Norman

Keys2: [maid, honor, matron, sister, wedding, guests, family, groom, bertha, father, norman]

All new keys: [bride, maid, honor, matron, sister, wedding, guests, family, groom, bertha, father, norman]

<<<<<<1st ROUND>>>>>>

Original Query: [attended, meloni]

word 0: attended

word 1: meloni

PERSONAL:

The bride is Meloni

Keys1: [bride]

<<<<<<2nd ROUND>>>>>>

word 0: bride

OPENMIND:

The maid of honor or matron of honor is the sister or best friend of the bride

OPENMIND:

The wedding guests are friends and family of the bride and groom

PERSONAL:

The bride is Meloni

PERSONAL:

The bride's mother is Bertha

PERSONAL:

The bride's father is Norman

Keys2: [maid, honor, matron, sister, wedding, guests, family, groom, bertha, father, norman]

All new keys: [bride, maid, honor, matron, sister, wedding,
 guests, family, groom, berth, father, norman]

<<<<<1st ROUND>>>>>

Original Query: [meloni, wedding]

word 0: meloni

PERSONAL:

The bride is Meloni

word 1: wedding

OPENMIND:

The wedding guests are friends and family of the bride and groom

OPENMIND:

The wedding guests are served slices of wedding cake

PERSONAL:

The groom's father is not present at the wedding

Keys1: [guests, family, bride, groom, wedding, guests, served, slices, cake, father]

<<<<<<2nd ROUND>>>>>>

word 0: guests

OPENMIND:

The wedding guests are friends and family of the bride and groom

OPENMIND:

The wedding guests are served slices of wedding cake

word 1: family

OPENMIND:

Groomsmen are male friends or family of the groom

OPENMIND:

The wedding guests are friends and family of the bride and groom

word 2: bride

OPENMIND:

The maid of honor or matron of honor is the sister or best friend of the bride

OPENMIND:

The wedding guests are friends and family of the bride and groom

PERSONAL:

The bride is Meloni

PERSONAL:

The bride's mother is Bertha

PERSONAL:

The bride's father is Norman

word 3: groom

OPENMIND:

Groomsmen are male friends or family of the groom

OPENMIND:

The best man is usually the brother or best friend of the groom

OPENMIND:

The wedding guests are friends and family of the bride and groom

PERSONAL:

The groom is Mantell

PERSONAL:

The groom's mother is Rose

PERSONAL:

The groom's father is not present at the wedding

word 4: wedding

OPENMIND:

The wedding guests are friends and family of the bride and groom

OPENMIND:

The wedding guests are served slices of wedding cake

PERSONAL:

The groom's father is not present at the wedding

word 5: served

OPENMIND:

The wedding guests are served slices of wedding cake

word 6: slices

OPENMIND:

The wedding guests are served slices of wedding cake

word 7: cake

OPENMIND:

The wedding guests are served slices of wedding cake

word 8: father

PERSONAL:

The bride's father is Norman

PERSONAL:

The groom's father is not present at the wedding

Keys2: [groomsmen, male, maid, honor, matron, sister, bertha, norman, brother, mantell, rose]

All new keys: [guests, family, bride, groom, wedding, served, slices, cake, father, groomsmen, male, maid, honor, matron, sister, bertha, norman, brother, mantell, rose]

3521-96