Semantic Understanding and Commonsense Reasoning in an Adaptive Photo Agent

by

Xinyu Hugo Liu

B.S., Computer Science and Engineering (2001)

Massachusetts Institute of Technology

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering In Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

May 2002

Signature of Author ……...
Department of Electrical Engineering and Computer Science
May 24, 2002

Certified by ……...
Dr. Henry Lieberman
Research Scientist, MIT Media Lab
Thesis Supervisor

Accepted by ……..
Arthur C. Smith
Chairman, Department Committee on Graduate Theses
Department of Electrical Engineering and Computer Science

Semantic Understanding and Commonsense Reasoning in an Adaptive Photo Agent

by
Xinyu Hugo Liu

Submitted to the Department of Electrical Engineering and Computer Science

May 24, 2002

in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering In Electrical Engineering and Computer Science

ABSTRACT

In a story telling authoring task, an author often wants to set up meaningful connections between different media, such as between a text and photographs. To facilitate this task, it is helpful to have a software agent dynamically adapt the presentation of a media database to the user's authoring activities, and look for opportunities for annotation and retrieval. Expecting the user to manually annotate photos with keywords greatly burdens the user. Furthermore, even when photos are properly annotated, their retrieval is often very brittle because semantic connections between annotations and the story text that are "obvious" to people (e.g. between “bride” and “wedding”) may easily be missed by the computer.

ARIA (Annotation and Retrieval Integration Agent) is a software agent that acts as an assistant to a user writing e-mail or Web pages. As the user types a story, it does continuous retrieval and ranking on a photo database. It can use descriptions in the story text to semi-automatically annotate pictures based on how they are used. The focus of this thesis is threefold: Improving ARIA’s automated annotation capabilities through world-aware semantic understanding of the text; making photo retrieval more robust by using a commonsense knowledge base, Open Mind Commonsense, to make semantic connections between the story text and annotations (e.g. connect "bride" and "wedding"); and learning personal commonsense through the text (e.g. “My sister’s name is Mary.”) that can then be used to improve photo retrieval by enabling personalized semantic connections.

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

Contents

Acknowledgements

This work would not have been possible without the help of many important people.

First and foremost, I would like to thank my thesis advisor and mentor, Henry Lieberman, for giving me the opportunity to work with him on such a wonderful and important project. His perspective on combining Artificial Intelligence with Software Agents has made a great impression on my own research interests. I thank him for showing me the enlightened path to effective research, for all the encouragement, direction, and flexibility that he has given me, and for his friendship.

Second, I would like to thank Push Singh, my one-time supervisor and current collaborator, for showing me the doorway to the world of commonsense reasoning. Without his important project, Open Mind Commonsense, my thesis work would literally not be possible. I want to thank him for over two years of enlightened conversations, close collaboration, and valued friendship. Push's wonderful talent for looking at the big picture in AI has been a great inspiration to me. Thanks for showing me that it can be both safe and fun swimming at the deep end of the AI pool.

Third, I want to thank Marvin Minsky, who has greatly inspired me with his forward-looking ideas like Society of Mind, and Emotion Machine. His work is why I wanted to do AI in the first place. I deeply appreciate his personal advocacy and support of my future research aspirations.

Fourth, I want to thank Bob Berwick, Ted Selker, and Walter Bender for the different ways in which they have helped to support my future research aspirations. Thanks for your faith in me.

Fifth, I want to thank my officemate Earl Wagner for his friendship and his ideas. Being at ground zero in the office, he was often the world's first line of defense against my irrational exuberance and my outlandish ideas. His very rational criticisms have helped me to debug my own ideas.

Sixth, I want to thank my mom and my dad for all their love and encouragement. Thanks mom, for calling me and reminding me that I also need to eat and sleep. Thanks dad, for giving me advice with perspective, and always being ready and willing to proof read my papers.

Lastly, I want to thank God for all of his blessings, for seeing me through the good times and the bad, and for giving me the greatest perspective on life.

Chapter 1

Introduction

The role of computers in society is rapidly changing. Applications of computers are so pervasively linked with the real world, that, increasingly, computer programs emulate human agents to teach, advise, and assist people in various tasks. The evolution of traditional computer programs into human-like software agents is already underway. We argue that in order for such a transformation to be successful, the traditional computer program will need to be upgraded in three important ways. 1) It should interact with the user in a more natural way, such as through human language in speech or text form. 2) It should be able to exercise some human "common sense" and have a basic awareness of the everyday world in which it operates. 3) It should learn from its interactions with the user and improve its interactions over time.

Human-like software agents do not yet exist. However, progress being made on the three fronts of natural language, commonsense reasoning, and user learning can be applied to design incrementally better software agents. This thesis describes how ideas from natural language, commonsense reasoning, and user learning have been incorporated into a software agent called ARIA, which assists users in organizing and using digital photos to tell stories through email and web page authoring. We show how the synergy of these three technologies in ARIA has made it more useful, adaptive, and

intelligent, and we offer this new ARIA as an example of how the gap from traditional computer program to full human-like software agents might be bridged. Before we present the ARIA photo agent, we motivate the project with some background on the traditional challenges of organizing and retrieving digital photos.

1.1 Motivation

As digital photography becomes more popular, consumers will need better ways to organize and search their large collections of images accumulated over time. The organization and retrieval of photos is a non-trivial task. Ideally, machine vision and image recognition will be able to automatically "understand" photos and photos would be able to organize themselves and be easily retrievable. Unfortunately, that technology is still far beyond us. Until then, most viable approaches are grounded in textual keywords.

Naïve Approach

Without any special software to assist them, users organize photos using the operating system. A common approach is to rename the image file from its default nonsensical name, something like "DCP00001.jpg," to something more meaningful like "graduation1.jpg." These files are then usually filed away into file folders. In fact, this hierarchical organization of photos into directories and albums is also the most common method of organization used by software programs such as Kodak's Picture Easy, which purport to help users organize photos.

The hierarchical approach has several drawbacks. First, there is an upfront cost imposed on the user for naming every photo and for creating hierarchies to house them. Second, photos must be put under one and only one folder or category. For example, the

category may be a place, social group, time, or event. The problem is that most photos naturally fall under many categories, so it may be hard for the user to choose how to organize the photos. Third, it may be difficult to retrieve photos, because at the moment of retrieval, the user may have in mind a different organizational method (e.g. place, social group, time, event) than the actual one, and a different name for the picture in mind than the actual name.

Using Annotations

A better way to organize photos may be through annotations. An annotation is defined as a keyword or phrase associated with a photo. Each photo can have multiple annotations. Whereas hierarchical structure restricts photos to one type of classification such as event, geographical location, etc., multiple annotations allow for multiple classifications. While annotation seems to solve the multiple-classification problem, two more problems remain. First, manually annotating photos is still a tedious task for a user, and in particular, it can be hard to choose the proper keywords, and to keep a consistent standard for keyword selection. Second, even with a collection of properly annotated photos, retrieval is only possible if the exact keywords that were used to annotate the photos are later recalled.

To make annotations an effective organizational method, the annotation process needs to be efficient, preferably somewhat automated, and retrieval needs to be much more robust than strictly keyword matching. Furthermore, to make it easier for the user to accomplish common tasks with photos, such as telling stories with photos in emails and web pages, both annotation and retrieval should be integrated into the user's task.

The ARIA Photo Agent attempts to integrate semi-automatic annotation and robust retrieval into a photo storytelling task.

1.2 The ARIA Photo Agent

In a story telling authoring task, an author often wants to set up meaningful connections between different media, such as between a text and photographs. To facilitate this task, it is helpful to have a software agent dynamically adapt the presentation of a media database to the user's authoring activities, and look for opportunities for annotation and retrieval. Expecting the user to manually annotate photos greatly burdens the user, and even if annotations do exist, potential connections to the story text are often missed because of differences in vocabulary or semantic connections that are "obvious" to people but that might not be explicit.

ARIA (Annotation and Retrieval Integration Agent) is a software agent that acts as an assistant to a user writing e-mail or Web pages. As the user types a story, it does continuous retrieval and ranking on a photo database. It can use descriptions in the story to semi-automatically annotate pictures based on how they are used in the text. To improve the associations beyond simple keyword matching, we heuristically identify text relevant to a photo, and perform world-aware parsing to extract important roles played by text, such as "who, what, where, when". Since many of the photos depict common everyday situations such as weddings or recitals, we use a commonsense knowledge base, Open Mind Commonsense (Singh, 2002), to fill in semantic gaps between the text and annotations (e.g. connect "bride" and "wedding") that might otherwise prevent successful associations. ARIA also attempts to learn personal commonsense that the user might

possess, such as "My sister's name is Mary," from its interactions with the user. Personal commonsense, just like our ordinary notions of "common sense," is used to make retrieval more robust.

A Brief History

In 1998, Lieberman et al. introduced a basic version of ARIA (2001), which was in and of itself a major innovation because it integrated the tasks of annotation and retrieval, and automated these tasks somewhat through observational learning of the user's behavior in typing an email. In this early version, automated annotation was done through naïve keyword association, and retrieval was also purely keyword based.

The work of this thesis is to help ARIA realize its full potential. ARIA's automated annotation capabilities are improved with semantic understanding of the text through natural language processing techniques. With a new concept-based knowledge representation for annotations, and through many modes of commonsense reasoning, retrieval is made robust. Temporal processing allowed for implicit queries over pictures within a time interval. A personal commonsense learner is able to learn from the user's interactions with the agent, in order to improve retrieval. The descriptions of ARIA presented henceforth reflect a post-thesis version of the system.

In the rest of this section, we briefly describe the user interface to ARIA and give a scenario of how ARIA can be used. The rest of the chapter discusses three specific research goals of this thesis, and then presents how these goals were manifested in the ARIA system architecture.

1.2.1 User Interface

Figure 1 shows the layout of the user interface. The text pane on the left is purposed for both emails as well as simple web page authoring. As the user types an email or web page, the photos in the photo database in the right pane dynamically re-rank and present themselves in order of anticipated relevance to what the user is typing. An index of each photo is shown in the photo pane, along with annotations for the photo, organized into "Who," "What," and "Where." (The "When" is implicit by the photo's timestamp).

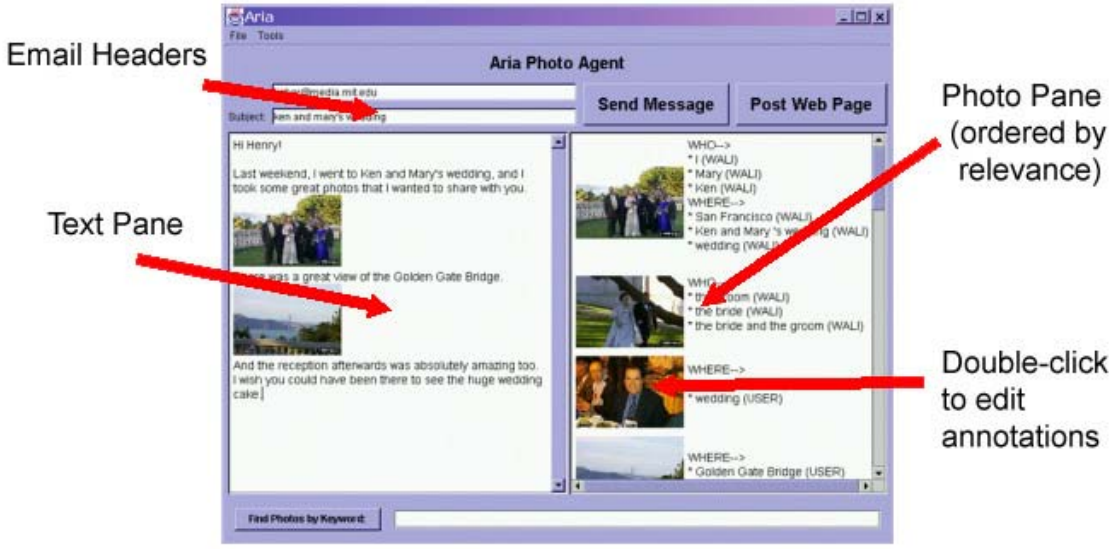

Figure 1: The layout of the ARIA user interface.

The user can include a photo in the text by dragging and dropping it into the desired position in the text pane. Although annotations are automatically learned when photos are used in the story, they may also be manually added and edited if a user double-clicks on a photo, as shown in Figure 2. In the Annotation Editor dialog box, annotations can be added under three metadata fields: WHO, WHAT, and WHERE. The COMMONSENSE field only appears in this debugging version.

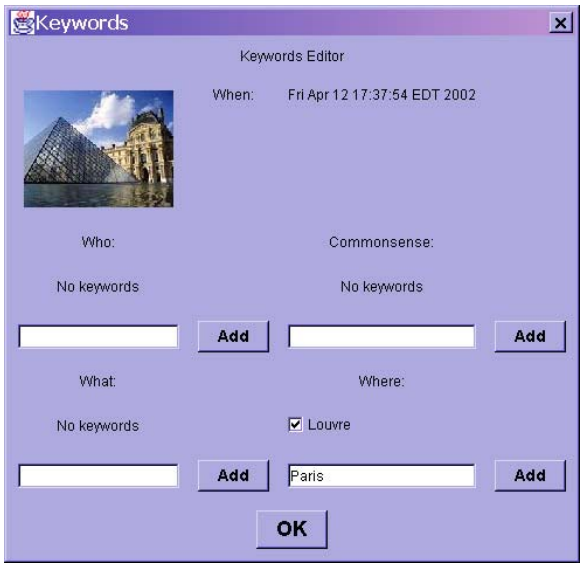

Figure 2: Annotation Editor Dialog.

Retrieval of photos is automatic as the user writes the story; however, photos can also be manually retrieved by entering keywords into the white text box at the bottom of the screen. When the story is completed, the user can fill out the email headers at the top of the screen and hit "send message" to send the email, or alternatively, the user can hit "post web page."

The commonsense used to connect the text and photos in a robust way, as well as any personal commonsense learned from the user's texts over time, are not shown or accessible to the user. Before we proceed to examine the annotation, retrieval, and learning mechanisms, we provide a comprehensive example of a user's interactions with ARIA to illustrate the full-range of the system's capabilities.

1.2.2 A Scenario

ARIA assists the user in her authoring task by annotating photos with descriptions from the story, and dynamically adapting the presentation of the photos while the user writes the story.

We present an example here to illustrate the system's core functionality, the capabilities of automated annotation and the personal commonsense learner, and also the different forms of commonsense reasoning that ARIA is capable of in its presentation of photos to the user. Because the user-to-agent interactions are dynamic in nature, we will describe the interactions in the form of a script. First, we describe the situation.

The User's Situation

Our hypothetical user, Kim, has been using ARIA for a few months; so she has a few months worth of pictures in ARIA's shoebox, including her trip to Europe, pictures of her friends at the MIT Media Lab where she works, and pictures of an office party. She has used ARIA to post one or two photo web pages in the past, so some annotations have already been learned. She also went ahead and added a few annotations manually. In this scenario, Kim recently came back from San Francisco, where she attended her friend Mary's wedding. She decides to write a photo email to her friend Henry to share some of her experiences.

Interactions with ARIA

Kim inserts her digital camera flash card and starts ARIA. The UI displays, and without delay, the photos from the digital camera are automatically imported into ARIA's shoebox, and displays toward the top of the photo pane. Kim decides to add a few

annotations to a couple of pictures. Kim fills out Henry's email address and a subject, and proceeds to type the email. (Her text is shown in bold italicized and comments are in square brackets).

1) ***Hi Henry!*** [pictures annotated with Henry emerge at the top of the shoebox]

2) ***Last weekend,*** [pictures taken last weekend emerge] ***I went to Ken and Mary's*** [pictures of Ken and Mary emerge] ***wedding in San Francisco.*** [Kim drags a group photo of attendees of the wedding from the top position in the shoebox into the email. As she does so, the following new annotations are automatically added to that photo: WHO→ Mary, Ken, Me; WHERE→ San Francisco, Ken and Mary's Wedding, wedding. Kim continues typing.]

3) ***Here is a picture of me*** [pictures of Kim in Paris and London emerges] ***and my officemate Mark, who also attended.*** [Kim drags in the picture of the two of them. In doing so, the photo gets annotated with: WHO→Mark, Me. Additionally, though not shown, the personal commonsense learner has learned the fact, *I have an officemate named Mark.*]

4) ***The patio offered a great view of the Golden Gate Bridge*** [As this is typed, some pictures that Kim had annotated with "San Francisco" come up, including one of the Golden Gate. ARIA made the association between San Francisco and the Golden Gate Bridge. Kim drags in a photo of the Golden Gate taken at the wedding. The photo now has the annotations: WHERE→ San Francisco, Golden Gate Bridge. This is an example of an interaction where ARIA guessed that the photo might have contained the Golden Gate based on its previous annotation of "San Francisco," and by using the photo, the user confirms that

the guess was correct, and so the annotation "Golden Gate Bridge" can be formally added to the photo.]

5) ***The ceremony*** [pictures annotated with "wedding" come up, because ARIA does a bit of social reasoning and knows that ceremonies are a component of weddings.] ***was magical and love*** [pictures with the annotations "bride" and "groom" come up, as well as pictures with the annotation "wedding." This illustrates some simple commonsense emotional reasoning.] ***was in the air.*** [Kim inserts a picture of Ken and Mary walking down the aisle. ARIA automatically annotates that photo with: WHERE→ ceremony.]

6) ***After the couple*** [a picture annotated with "bride" and "groom" appears.] ***took their vows, everyone headed for the reception.*** [a picture of the reception appears, though it had no annotations. How did ARIA know this? ARIA did a bit of temporal reasoning. It knew the commonsense fact that, *After a wedding ceremony there may be a reception,* and since another photo was annotated with "wedding," and *this* photo's timestamp was a few hours after *that* photo's timestamp, ARIA guessed correctly that this was a photo of the reception. Kim was surprised to see the photo, and though she wasn't originally planning on including photos of the reception, seeing this photo made her change her mind. This is an example of how ARIA and the user's authoring behavior are mutually adapting. She drags it in. That photo now gets annotated with: WHERE→ reception.]

7) ***In the middle of the reception, people decided to do a traditional Jewish dance. Here is a picture of the couple*** [photo annotated with "bride" and

"groom" pop up] ***doing the "hora".*** [Kim inserts a picture of the couple doing the hora, and the annotation "hora" was learned. ARIA figured out that "hora" was relevant enough to the photo to make it an annotation, even though it has no knowledge of what "hora" means. It did this by inferring the semantic importance of the word based on its syntactic usage.]

8) ***I haven't had that much fun since our office party.*** [a picture of the office party with the annotations "Mark," "Brian," and "cocktails" emerges. ARIA put together several pieces of partial evidence. From personal commonsense learned earlier in the interaction, it knew that "Mark" was the user's officemate, and from commonsense, it associates officemates with offices. And it had some commonsense that *People drink cocktails at parties,* so it associated the annotation "cocktails" with "party."]

9) ***Cheers, Kim.*** [Kim clicks send and ARIA sends the photo email in HTML to Henry.] END OF SCENARIO

We give a brief review of the points illustrated by the above scenario. First, ARIA suggested photos which were possibly relevant to the text of the story, in a non-invasive way. When Kim found a suggestion to be useful, she incorporated it; however, when suggestions were not particularly relevant, she could continue on with her task without much distraction. This highlights the fail-soft nature of photo retrieval in ARIA.

Second, through a combination of matching keywords in the story text to annotations, matching time intervals, and guessing photos with possible semantic relevance, ARIA is able to provide fairly robust retrieval. ARIA is able to parse temporal phrases like "last

weekend," and use that information to locate photos taken in a particular time interval. To fill in the semantic gaps between words like "bride" and "wedding," ARIA uses some commonsense reasoning. In the scenario, ARIA was able to make several kinds of commonsense associations, including 1) *Geospatial*, i.e. "golden gate bridge" with "San Francisco"; 2) *Emotional*, i.e. "love" with "bride", "groom", and "wedding"; 3) *Temporal*, i.e. "wedding ceremony" with "wedding reception", using photo timestamps; 4) *Social*, i.e. "cocktails" with "party"; and 5) *Personal*, i.e.: "Mark," with "office."

Third, ARIA was able to automatically learn some annotations, and classify the annotations under the categories: Who, What, and Where. It was able to judge what is worth annotating, versus what is not, and when given unknown words like "hora," it was able to guess the semantic relevance of that concept to the photo, based on structural cues.

Fourth, ARIA observed the user's text and learned pieces of personal commonsense knowledge. For example, ARIA remembered that "Mark" was Kim's "officemate," and added that to its user model. Later on in the scenario, that piece of knowledge was used to connect "office party" to a picture of that event annotated with "Mark."

In the following section, we put the annotation, retrieval, and learning mechanisms illustrated by this scenario into the perspective of the research objectives of this thesis.

1.3 Research Objectives

At the very beginning of this thesis, we suggested that three salient properties would distinguish intelligent software agents from ordinary programs. An intelligent software agent should 1) interact with the user in a more natural way, such as through human

language in speech or text form; 2) be able to exercise some human "common sense" and have a basic awareness of the everyday world in which it operates; and should 3) learn from its interactions with the user and improve its interactions over time.

The research objectives of this thesis are to investigate how these three properties can be endowed upon ARIA to make it a more intelligent and adaptive software agent. These broad properties map into more concrete research objectives as follows.

With respect to ARIA, the implication of interacting with the user through natural language needs careful consideration. ARIA is by nature an assistive agent, and it merely facilitates the user's task of sending photo emails. It does not need to conduct dialog with the user, but if it is to behave intelligently, it should assist the user with annotation and retrieval. Because annotation and retrieval are contextualized within the scope of the user's task, ARIA will need a deep level of understanding of the natural language text. ARIA should know enough about the syntactic and semantic structures of the text to know how to automatically annotate photos with descriptions from the text. We will refer to this as *Observational Learning of Annotations*, to be achieved through semantic understanding of the text.

The second goal of giving the software agent some commonsense and world-awareness to help its task is more straightforward. In thinking about how ARIA might need commonsense, we are reminded of the problem of the retrieval mechanism being very brittle. While a person might find the relationship between a "bride" and a "wedding" to be obvious, computer programs do not. How might the retrieval be made more robust? We make the observation that photos are snapshots of the real world, and therefore, knowing the established commonsense relationships between people, places,

things, and events in the everyday world can help the retrieval mechanism more intelligently associate concepts that are *world semantically related.* We will refer to this second research objective as *Robust Photo Retrieval*.

The third property we wish ARIA to have is that it should learn from its interactions with the user over time. The system can learn many things from its interactions, but one of the most useful things ARIA can learn is the user model, that is to say, details about the user such as friends, family, etc. that are obvious or commonsensical to the user. Knowing such details might help enhance retrieval. For example, if a picture is annotated with the name of the user's sister, "Jane," a goal might be to allow the user to retrieve that picture by typing "my sister." This third research objective will be referred to as *Learning from and Adapting to the User Model.*

We elaborate each of the three research objectives in the following subsections. Observational learning of annotations in Section 1.3.1; Robust photo retrieval in Section 1.3.2; and Learning from and Adapting to the User Model in Section 1.3.3.

1.3.1 Observational Learning of Annotations

One of the more innovative aspects of ARIA is that it unloads some of the burden of annotation off the shoulders of the user. While the user performs the task that s/he would do so anyway, ARIA learns annotations based on how they are used in the text. Portions of the text relevant to a particular photo can be identified, and the relevant annotations can be extracted out of descriptions in the text. However, it is non-trivial to devise a mechanism that can perform this task competently and consistently.

Computer visions algorithms are not good enough to "see" the subjects of the photos, so all that a computer program can do is to make intelligent guesses as to what the annotations should be, from clues and descriptions in the text. Sometimes, there are textual cues, such as "Here is a picture of …," and heuristics *can and should* be devised to take advantage of such knowledge; however, if such cues are not present, the annotation mechanism needs to figure out what parts of the text may be relevant to the photo and should be included as an annotation. In order to pick out what is relevant from the text, the mechanism will need to have some understanding of the text insofar as what is important, and what is referring to the picture. Figure 3 depicts a simple example, which we will use to evaluate different approaches.

Hi Henry!

Last weekend, I went to Ken and Mary's wedding in San Francisco. Here's a picture of the two of them at the gazebo.

Figure 3: An illustrative example of annotation

A Baseline Approach

Given this simple example, the simplest approach would be to extract all keywords from the text immediately preceding the photo and make all of them annotations. We will refer to this as the baseline approach, because it can be used to evaluate the

performance of other approaches. Applying the baseline approach to the example in Figure 3 would produce the following annotations:

Henry, Last, Weekend, Went, Ken, Mary's, Wedding,

San, Francisco, Picture, Gazebo.

As is routinely done with keyword spotting, we exclude a list of stop words such as determiners (the, a), prepositions (in, of, at, etc.), among others. At first glance, the baseline approach seems to be adequate. But there are several issues.

First, when some concepts are broken up into words, their meanings, or *semantics*, change. This includes phrases like "last weekend," and "San Francisco." In general, semantics is regarded as compositional, sometimes known as Montague semantics (Dowty, Wall, Peters, 1981), and the meanings of sentences are some composition of the meanings of the concepts contained. However, phrases at the concept level cannot be further decomposed. One of the problems of keyword annotations is that the annotations "Golden," "Gate," and "Bridge" will match things with much broader semantics than "Golden Gate Bridge," thus such annotations could prove to be somewhat misleading and inaccurate, in terms of the retrieval that results.

Second, the semantics of some words are dependent on place and time. For example, "last weekend," and more specifically, "last," are not useful annotations because their meanings change as the reference point (the current point in time) changes. This is more reason why words need to be treated as the concepts that they represent rather than as merely symbols.

Third, the baseline approach is not able to discern which keywords actually refer to the subject of the photo and which are important enough to include as annotations. The

annotation "Henry" is clearly irrelevant to the photo, and the annotation "picture" clearly does not refer to the subject of the photo. However, without a deeper understanding of the text, it is difficult to eliminate such annotations.

Fourth, the baseline approach does not know enough to determine the scope of knowledge that may be relevant to the photo. In the sample keywords given above, keywords were taken from all the sentences preceding the picture. However, one can easily imagine a scenario where many irrelevant sentences are placed before the photo, and hence, many inappropriate annotations would result. One way to solve this would be to only include the sentence immediately before the photo, but this could potentially exclude relevant annotations. For example, in the example of Figure 3, the only annotations would be **Picture** and **Gazebo.** It is clear that without more understanding of the text, one cannot hope to resolve any of the aforementioned problems of the baseline approach.

Syntactic Parsing

In order to achieve more understanding of the text, to be able to identify relevant passages in the text, and to extract relevant annotations, more sophisticated natural language processing is necessary. Syntactic parsers have traditionally been very popular for natural language processing tasks. Syntactic parsers output what is known as constituent structure trees, known as c-structures. Given a tokenized sentence like "I went to Ken and Mary's wedding in San Francisco," a syntactic parse tree is produced, resembling something like this:

```
[S
 [NP I]
 [VP
  [V went]
  [PP to
    [NP
      [NP [NP [NP Ken] and [NP Mary 's]] wedding]
      [PP in
        [NP San Francisco]]]]]]
```

Several syntactic parsers have been developed. Those that are publicly available include Link Grammar Parser (Sleator and Temperley, 1993), XTAG (XTAG Research Group, 1999), and ThoughtTreasure (Mueller, 1998a). Although syntactic parses provide hints as to the semantics of concepts and how they are related, they do not give a detailed enough picture. Our task of automated annotation would require an understanding of the *semantic type* (i.e. a categorization like `person`, `place`) of the concepts, and some measure of the semantic relevance of a concept to a photo.

Semantic Parsing

One possibility is to apply a semantic parser rather than a syntactic parser. Semantic parsers take in tokenized sentences or syntactic parse trees and output semantic representations such as frames and logical assertions. For the input sentence given earlier, the Cyc-NL semantic parser (Burns and Davis, 1999) might produce something like this:

```
(and (isa I1 Person)
     (isa Ken1 Person)
     (isa Mary1 Person)
     (isa San_Francisco1 City)
     (isa Event1 WeddingEvent)
     (hasBride Event1 Mary1)
     (hasGroom Event1 Ken1)
     (atLocation Event1 San_Francisco1)
     (isa Action1 GoingToAnEvent)
     (performedBy Action1 I1)
     (toEvent Action1 Event1))
```

This is not the actual output of the parser, but it is in the style of Cyc-NL and is used to illustrate a point. The Cyc-NL parser maps concepts in sentences into carefully described objects and frames encoded into Cyc, such the `WeddingEvent` frame and the `GoingToAnEvent` action frame. Though the level of detail provided by this logical formula is superior, this cannot be considered to be a broad coverage or robust parser, as it is unlikely that there are encodings for the breadth of concepts and relations we wish to cover.

While Cyc-NL tries to capture meaning using a "stamp collection" of abstractions, frames, and concepts, the UNITRAN parser (Dorr, 1992) reduces all concepts and actions into a small set of primitive operations. For the same input sentence, UNITRAN might output something like:

```
[GO_Ident
  ([Thing I],
   [TOWARD_Ident
        ([AT_Ident
            ([Thing I],
             [Location KEN AND MARY'S WEDDING])])])]
```

The primitives used by UNITRAN are an extension of Lexical Conceptual Structures (LCS), which is based primarily on work by Jackendoff (1983, 1990), and further studied by Zubizaretta (1982, 1987), Hale and Laughren (1983), Levin and Rappaport (1986), Hale and Keyser (1986a, 1986b, 1989). LCS is particularly well suited to describing spatial orientation and movement, i.e. paths, and also causality. In trying to identify the subjects of photos for the purpose of enumerating annotations, it may be useful to express sentences using an event ontology such as LCS. Understanding the movement and placement of the photographer and subjects in a photo may lead to a better guess as to the subject of the photo. There are, however, also many limitations to the approach.

Because all concepts must be boiled down into very few primitives, creating lexical entries could be a tedious process. Though UNITRAN may be a suitable representation for interlinguas, used in machine translation, it would likely be very difficult to construct a lexicon extensive enough to allow broad coverage parsing.

Though these semantic parsers would provide a deep level of understanding, they generally have inadequate coverage and robustness for domain-independent texts. The main reason for these shortcomings is that semantic parsers require extensive lexical knowledge such as detailed verb-argument structure, selectional preferences, thematic roles, etc. However, such resources are scarce and incomprehensive.

The few resources that do exist are WordNet, Comlex, Corelex, Levin Alternation classes, and FrameNet. WordNet (Fellbaum, 1998) disambiguates lexical items into word senses, associates each word sense with one of 34 argument structure patterns, and provides nymic relations for each word sense. Comlex (Grishman, Macleod, and Meyers, 1994) syntactic information includes argument structure for 38,000 English words. Corelex (Buitelaar, 1998) provides an ontology and semantic database of 126 semantic types for 40,000 nouns, and defines systematic polysemous classes to group words based on sense distribution data. Levin's English verb classes and alternations (1993) classify 4,183 verbs into 191 classes based on semantic similarity and syntactic structure. FrameNet (Baker, Fillmore, and Lowe, 1998) provides the frame semantics (Fillmore, 1968) for 2,000 lexical items (in FrameNet I) in a variety of discourse domains.

Parsers that provide deeper understanding will also face challenges with robustness because ARIA's text is in the particularly informal email domain. Text in this domain

tends to omit capitalization, punctuation, and words, and is often referred to as *mildly ill-formed.* In our task, it is important to balance understanding with robustness.

We briefly review the various NLP approaches discussed above. Syntactic parsers are broader in coverage and more robust than semantic parsers like UNITRAN and Cyc-NL. However, they do not contain semantic type information (i.e. Cyc-NL), and event structure (i.e. UNITRAN), both of which are useful to our task. Furthermore, though the depth of understanding provided by UNITRAN and Cyc-NL may be important to a particular class of problems, it is not as vitally important to our task of automating annotation.

A Hybrid Approach

We propose a hybrid solution of syntactic analysis, semantic typing, and event structure to solve the task of automated annotation. Such a solution should have the broad coverage and robustness properties of syntactic processing, and should not require the manual development of substantial semantic resources, which is still a major hurdle in deep semantic understanding. The cost of parsing and semantic processing should be justified by improved annotation accuracy and precision over more shallow techniques such as noun chunking and keyword spotting.

The proposed semantic understanding mechanism for automating annotation will consist of these general processing steps:

1) Syntactic Analysis of Text
2) World Semantic Reformulation
3) Relevance Identification
4) Annotation Extraction

First, a syntactic analysis of the text will be performed, including tasks such as tokenization, normalization, tagging part of speech, and syntactic constituent parsing. Continuing with the example given earlier, the sentence "I went to Ken and Mary's wedding in San Francisco" will produce the following syntactic structure. Part-of-speech tags are denoted by "/" followed by the name of the part-of-speech, according to the Penn Treebank tagset (Marcus, Santorini, and Marcinkiewicz, 1993).

```
[S
 [NP I/PRP]
 [VP
  [V went/VBD]
  [PP to/TO
    [NP
      [NP [NP [NP Ken/NNP] and/CC [NP Mary/NNP 's/POS]] wedding/NN]
      [PP in/TO
        [NP San/NN Francisco/NN]]]]]]
```

In the second step of processing, world semantic knowledge will be applied to the syntactic parse tree from the previous step to *transform* the syntactic constituent structure tags into event structure tags. By *world semantic knowledge*, we mean the knowledge about the semantic types of concepts grounded in the real world. Some examples of semantic types the mechanism may assign are as follows: "San Francisco" is a city, "wedding" is a social event, "anger" is an emotion, "Mark A. Smith" is a male person, "last weekend" is a time interval, etc. Such knowledge might come from a knowledge base of commonsense, because semantic type identification is a component of human commonsense. By *event structure tags*, we mean tags based loosely on Lexical Conceptual Structure. Tags derive from a small set of ontological categories focused around paths and motion (often thought of as conceptual parts of speech) such as *thing, event, state, action, trajectory, place, property, purpose, manner, amount, time.* A more

detailed review of LCS will take place in Chapter 2. After this step of processing, a structure such as the following may be produced:

```
[ASSERTION
 [PERSON I]
 [ACTION go
  [PATH to
    [EVENT
      [EVENT [PEOPLE [PERSON Ken] and [PERSON Mary/NNP 's]] wedding]
      [LOCATION in
        [PLACE San Francisco]]]]]]
```

The third step of processing concerns relevance identification, that is to say, given a body of text and a placeholder of a photo, how can the sentences and clauses relevant to the photo be identified? We propose a heuristic approach, which performs a weighted evaluation over many features such as structure, word cues, proximity, and also uses anaphor-to-referent placement as a salient feature. With this last feature, given a sentence which is considered relevant, and contains an anaphor, then the location sentence, which contains the referent for that anaphor, is also likely to be relevant. By this procedure, sentences can be "attached" to the relevance context space of a photo.

The last step of processing is annotation extraction. From the set of semantically processed sentences in the relevance context space, we can extract from sentences semantic roles contained in the subject of the photo. The semantic roles depicted in a photo can be thought of as falling under the Who, What, When, and Where of the photo.

More discussion on the proposed approach and its implementation will take place in Chapter 2. We continue on to discuss the second main research objective of this thesis.

1.3.2 Robust Photo Retrieval

One of the important tasks of this thesis is the robust retrieval of annotated photos by a keyword query. By "annotated photos," we mean a photo accompanied by some metadata about the photo, such as keywords and phrases describing people, things, places, and activities depicted in the photo. By "robust retrieval," we mean that photos should be retrievable not just by the explicit keywords in the annotation, but also by other implicit keywords obviously related to the subject of the photo.

The baseline approach for photo retrieval involves *exact keyword matching.* This approach is not robust because it is grounded in keywords, rather than in concepts. Three basic problems may hinder successful retrieval. 1) Morphological differences, as in the difference between *sister* and *sisters*, is one problem, though it is easily solved through word stemming and lemmatization (Ritchie et al., 1992; Sanfilippo, 1990; Karttunen et al., 1992). 2) Vocabulary differences, as in the difference between *car* and *automobile,* can also prevent successful retrieval, though it is also easily solved through lexical resources such as thesauri. 3) Semantically connected concepts, as in the similarity of *bride* and *wedding*, is perhaps the most difficult problem to tackle, because it requires some understanding of the connectedness of concepts in our everyday world. Our research objective is to make retrieval more robust by solving the *semantic connectedness* retrieval problem. In this section, we will first present common approaches to robust retrieval from the literature of information retrieval, and discuss their appropriateness to our task. We then propose applying some basic commonsense reasoning as a way to solve the problem.

Common IR Approaches

Given the task of solving *semantic connectedness* in retrieval, we first consider some standard approaches to document retrieval. Photos annotated with textual keywords can be thought of as resembling documents, and querying for photos by keywords is akin to the information retrieval done by search engines. In fact, the common query enrichment techniques such as thesaurus-based keyword expansion developed for document retrieval may be applied to the photo retrieval domain without modification. However, keyword expansion using thesauri is limited in its usefulness because keywords expanded by their synonyms can still only retrieve documents directly related to the original keyword. Furthermore, naïve synonym expansion may actually contribute more noise to the query and negate what little benefit keyword expansion may add to the query, namely: if keywords cannot have their word sense disambiguated, then synonyms for all the word senses of a particular word may be used in the expansion, and this has the potential to retrieve many irrelevant documents.

Attempting to overcome the limited usefulness of keyword expansion by synonyms, various researchers have tried to use slightly more sophisticated resources for query expansion. These include dictionary-like resources such as lexical semantic relations (Voorhees, 1994), and keyword co-occurrence statistics (Peat and Willet, 1991; Lin, 1998), as well as resources generated dynamically through relevance feedback, like global document analysis (Xu and Croft, 1996), and collaborative concept-based expansion (Klink, 2001).

Although some of these approaches are promising, they share some of the same problems as naïve synonym expansion. Dictionary-like resources such as WordNet and

co-occurrence frequencies, although more sophisticated that just synonyms, still operate mostly on the word-level and suggest expansions that are lexically motivated rather than conceptually motivated. In the case of WordNet, lexical items are related through a very limited set of nymic relations. Relevance feedback, though somewhat more successful than dictionary approaches, requires additional iterations of user action and we cannot consider it fully automated retrieval, which makes it an inappropriate candidate for our task.

An Approach with Commonsense

With regard to our domain of photo retrieval, we make a key observation about the difference between photos and documents, and we exploit this difference to make photo retrieval more robust. We make the observation that photos taken by an ordinary person has more structure and is more predictable than the average document on the web, even though that structure may not be immediately evident. The contents of a typical document such as a web page are hard to predict, because there are too many types and genres of web pages and the content does not predictably follow a stereotyped structure. However, with typical photos, such as one found in your photo album, there is a more predictable structure. That is, the intended subject of photos often includes people and things in common social situations. Many of these situations depicted, such as weddings, vacations, sporting events, sightseeing, etc. are common to human experience, and therefore have a high level of predictability.

Take for example, a picture annotated with the keyword *"bride"*. Even without looking at the photo, a person may be able to successfully guess who else is in the photo, and what situation is being depicted. Commonsense would lead a person to reason that

brides are usually found at weddings, that people found around her may be the groom, the father of the bride, bridesmaids, that weddings may take place in a chapel or church, that there may be a wedding cake, walking down the aisle, and a wedding reception. Of course, commonsense cannot be used to predict the structure of most specialty photos such as artistic or highly specialized photos; the approach taken in this thesis is only immediately applicable in the realm of consumer photography.

Building and Using a World Semantic Resource

Knowledge about the spatial, temporal, and social relations of the everyday world is part of commonsense knowledge. We also call this *world semantics*, referring to the meaning of everyday concepts and how these concepts relate to each other in the world. The mechanism we propose for robust photo retrieval uses a world semantic resource in order to expand concepts in existing photo annotations with concepts that are, *inter alia,* spatially, temporally, and socially related. More specifically, we automatically constructed our resource from a corpus of English sentences about commonsense by first extracting predicate argument structures, and then compiling those structures into a Concept Node Graph, where the nodes are commonsense concepts, and the weighted edges represent commonsense relations. The graph is structured much like MindNet (Richardson et al., 1998). Performing concept expansion using the graph is modeled as spreading activation (Salton and Buckley, 1988). The relevance of a concept is measured as the semantic proximity between nodes on the graph, and is affected by the strength of the links between nodes.

The proposed approach is elaborated in Chapter 3, and an implementation is given. We proceed to discuss the third research objective, learning from and adapting to the

User Model. As we will see later, knowledge learned about the user can be thought of as a sort of "personal commonsense" because it is knowledge which is obvious to the user. As such, personal commonsense also fits aptly under the associative reasoning mechanism proposed for ordinary commonsense, which was presented in this section.

1.3.3 Learning from and Adapting to the User Model

In the previous section, we discussed how knowledge about how concepts relate to each other in the everyday world might be used to identify semantically related concepts. However, there is another type of association that can be made to improve retrieval – knowledge about the user that is obvious to the user. For example, in knowing that the name of the user's sister is Sally, it becomes possible to refer to both *Sally* and *my sister* interchangeably in the retrieval of a photo annotated with either string. In the literature of Human-Computer Interaction (HCI), this type of knowledge about the user may be referred to as part of a *user model*. Because of the striking similarities between how ordinary commonsense and personal knowledge is used, this knowledge may also be considered *personal commonsense.* In this thesis, we will use the terms u*ser model* and *personal commonsense* interchangeably, depending on which quality of the knowledge is being emphasized.

As suggested by the heading of this section, it is our goal both to learn the user model, and to have future interactions with the user adapt to the known user model, improving the system's behavior accordingly. The learning algorithm we propose uses the same mechanism for parsing of text into a shallow event structure presented earlier. Working with the output of the parser, we propose a rule-based approach to mapping the

parse tree into an ontology of relations we wish to capture in the realm of personal commonsense. The mapping rules considers some thematic role information, which is extracted from the parse tree, the nature of the event structure, and most importantly, the semantic types of concepts.

The ontology of relations captured should include basic relationships such, inter alia, as people's relationships to the user, e.g. *Mary is the name of my sister,* and people's relationships to places, e.g. *Bob is often found at church*. Without the capability of full human-level understanding of the text, learning associations between concepts will require a statistical approach. While some relationships can be learned after a single instance in the text with an acceptable certainty, other relationships may require multiple instances in the text, possibly over many different interactions, to be learned. The rules of the learner will generally be of two types: pattern-matching, and conceptual co-occurrence. Pattern-matching has always been a staple of text processing tools. Pattern match rules in our system provide templates for recognition of relations, which take advantage of salient structural features and semantic types. These rules generally correspond to relationships that can be learned after a single example. The main limitation of such rules are that they are rather sensitive to syntax, and verb alternations; however, pattern-matching rules are rather useful for common cases in which personal commonsense is introduced.

The second type of rule aims to learn relationships through *conceptual co-occurrence*. In the computational linguistics literature, co-occurrence, also known as collocation, generally refers to the occurrence of two keywords within some proximity of each other. Co-occurrence between words usually suggests either syntactic or semantic relatedness.

By *conceptual co-occurrence*, we simply mean that we will consider the co-occurrence of two semantically typed concepts, such as people with places. This type of rule generally corresponds to relationships that can only be learned after many examples. The main advantage of the conceptual co-occurrence approach is that it is not sensitive to the syntax and surface realization of the underlying meaning of the text, because no structural cues are considered.

By combining pattern-matching rules for common cases, and conceptual co-occurrence rules to cover broader cases, the proposed learner should be able to learn a broader range of relationships. Once personal commonsense is learned, the same mechanism proposed to apply ordinary commonsense to semantically expand annotations can be used by the system to adapt and improve its retrieval to the knowledge from the user model.

Chapter 4 provides a more in-depth discussion and an implementation of the proposed approach described here. The following section describes how our three research objectives might be integrated as components in the larger system, ARIA.

1.4 System Architecture

We preface the presentation of the system architecture by recapitulating our three research objectives. This thesis investigates the observational learning of photo annotations, achieved through world-aware constituent parsing of the text; making photo retrieval more robust by using commonsense reasoning to make semantic connections between related concepts; and observationally learning and adapting to the user model in order to improve the retrieval behavior of ARIA based on user-specific knowledge.

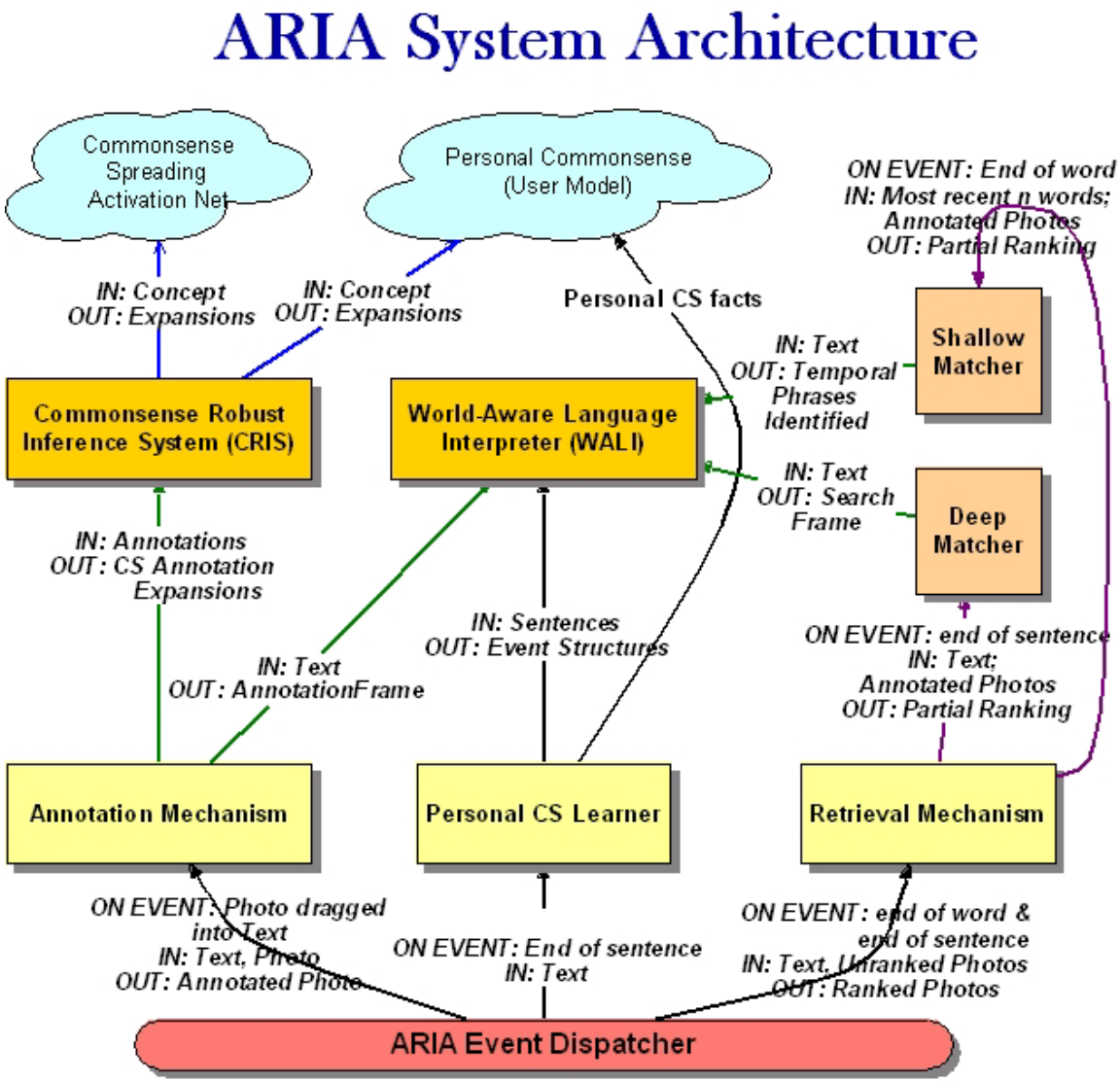

Figure 4: ARIA Backend Systems Architecture. GUI, text editing, email, and HTML components not shown.

As illustrated in Figure 4, these three research objectives manifest themselves as three mechanisms, for annotation, retrieval, and learning. In addition, two large subsystems, WALI and CRIS, are shared by, and provide services to, the three mechanisms. We now present an overview of the two subsystems, and then of the three mechanisms.

1.4.1 Subsystems Overview

The *World-Aware Language Interpreter* (WALI) is the semantic understanding subsystem which, when passed a body of text, can identify sentences relevant to a photo, can produce a world semantic event structure parse of the text, and can extract semantic roles of Who, What, When, and Where from the text. WALI provides annotation extraction services to the annotation mechanism, sentence understanding services to the Personal Commonsense Learner mechanism, and semantic role extraction and temporal phrase resolution services to the retrieval mechanism.

Commonsense Robust Inference System (CRIS) is the commonsense expansion subsystem, which, when passed a concept or annotation, can return a weighted list of semantically related concepts. It uses both ordinary commonsense as well as personal commonsense as its source of knowledge for the expansions. Before runtime, these two sources of knowledge are compiled from semi-structured English sentences into efficient concept graphs to allow quick exploration. The construction of these semantic resources is discussed in Chapter 3.

CRIS provides the concept expansion service to the annotation mechanism, so that annotations are expanded whenever a photo's annotations are modified. An alternative design choice would be to expand incoming words in the body of the text, and to match those against unexpanded annotations; but such an approach faces efficiency problems because photo suggestions are made on-the-fly, after each new word is typed. The commonsense expansion process is currently too computationally expensive for expansion on the retrieval side to make any sense.

These two main subsystems provide services to three main mechanisms for annotation, retrieval, and learning, which are presented below.

1.4.2 Mechanisms

Annotation

The annotation mechanism is dispatched when a photo is dragged from ARIA's photo shoebox into the text pane, and the entire contents of the text window, as well as a pointer to the photo are passed to it. Using the WALI subsystem, the sentences relevant to the picture are identified, the text is *understood* as world-semantic event structure, and the semantic roles of Who, What, When, and Where are extracted from the relevant text. We refer to the collection of concepts organized into semantic roles as an *Annotation Frame*. WALI passes back an annotation frame. If the particular photo already has annotations organized into annotation frame, the two frames are merged additively. Annotations in the merged annotation frame are then passed to the CRIS subsystem, where an attempt is made to expand each annotation using ordinary commonsense as well as personal commonsense. All annotation expansions, which meet some minimum threshold confidence score, are merged into the photo's annotation frame. Each annotation in the annotation frame has a confidence score associated with it, and that is figured into the retrieval process. Commonsense expansion annotations have lower confidence scores than annotations learned by ARIA, which in turn have lower confidences scores than annotations manually entered by the user. This makes sense because the purpose of commonsense expansions is to add robustness to the retrieval when no learned or manual annotation exists. Sometimes many low-confidence score annotations can combine to

retrieve a photo with high relevance to the user task. The phenomenon of combining partial evidence is the basis for retrieval, which we discuss next.

Retrieval

The retrieval mechanism is dispatched by ARIA on two events: word breaks and sentence breaks. Word breaks cause the invocation of the shallow retrieval mechanism, while sentence breaks invoke a deeper retrieval mechanism. Shallow retrieval allows photo suggestions to be made on a local three-word context of what the user types, while deeper retrieval extracts all possibly relevant concepts from the most recent sentence and uses a combination of these concepts to create photo suggestions that may be relevant to the entire sentence.

The shallow matcher component is called more frequently than the deep matcher, so it must be rather lightweight and efficient. The shallow matcher performs simple tokenization on keywords, lemmatization, and temporal phrases are resolved. After this processing, keywords are string matched to each annotated photo. Lemmatization makes string matching more robust to morphological variations, such as plurals. If there is a temporal phrase, the time interval corresponding to it is matched against the timestamp associated with the photo. By combining temporal and keyword evidence, a relevance score is produced for each photo in the ARIA shoebox, and a ranking is obtained.

Whereas the shallow matcher ranks photos based on a local three-word context, the deep matcher's suggestions take into account an entire sentence. After each sentence, the deep match uses WALI to extract semantic roles from the sentence, also organized into a frame, which we refer to as a Search Frame. A search frame is slightly different from an annotation frame because it may have a time interval associated with it. A scoring

function matches the annotations of the search frame against the annotation frame of each photo, and the time interval of the search frame against the timestamp of each photo. Similar to the shallow matcher, photos are scored based on the quality of the match, and a ranking is obtained.

The retrieval mechanism uses the rankings given of the shallow and deep matcher to reorder the photos in the ARIA shoebox such that the most relevant photo appears on top, second most relevant photo appears second, and so on. Even though the stated purpose of the commonsense is ARIA is to make photo retrieval more robust, no commonsense reasoning is invoked by the retrieval mechanism. Instead, the task is given to the annotation mechanism, for the sake of efficiency.

Personal Commonsense Learner

As we stated earlier in describing the CRIS subsystem, personal commonsense is used in the same way as ordinary commonsense to expand annotations. The stated purpose of this mechanism is to learn personal commonsense facts about the user, styled as semi-structured English sentences, through observational learning of the text. Called at the end of each sentence, the learner invokes WALI to generate a world-aware parse of the text, and then applies its pattern-matching and conceptual co-occurrence rules to learn relationships. It uses a persistent database to keep track of possible relationships it sees in its interactions with the user. When a piece of personal commonsense can finally be learned, it is reformulated as a semi-structured English sentence, and stored, for use by CRIS. Personal commonsense learned takes effect immediately by updating the expansions of all existing annotations using the newly learned fact.

Having presented the architecture of the system implemented for this thesis, the rest of the paper is organized as follows. Chapter 2 presents a discussion and an implementation of the semantic understanding subsystem we have proposed to solve the task of automated annotation. Chapter 3 presents a discussion and an implementation of the commonsense reasoning subsystem used to make retrieval more robust. The personal commonsense learner is discussed is Chapter 4. Chapter 5 presents related work, including contrasting approaches to photo management, and other software programs we have developed that have attempted to incorporate commonsense in some way. We conclude with a summary of the contributions this thesis makes, and suggest future work that might be done.

Chapter 2

Semantic Understanding for Automated Annotation

One of the main challenges in building a semantic understanding system is getting the knowledge representation right. The depth of understanding that can be achieved corresponds to the sophistication of the knowledge representation used. However, more sophisticated representations require more comprehensive and sophisticated language resources, such as lexicons, dictionaries, parts of speech, verb argument structures, selectional preferences, frame semantics, and world semantics. Many of these resources are not available, or are not complete, so deeper understanding in general implies more brittleness. The tradeoff between depth of understanding and robustness needs to be an important consideration when deciding on a knowledge representation. For understanding text, there are two well-known types of machinery that demonstrate the tradeoff well.

Syntactic processing and parsing achieves less understanding, but is acceptably robust. Given a tokenized sentence, a syntactic parser can perform noun phrase chunking, and elucidate some governance relationships between verbs, noun phrases, prepositional phrases, and sentence clauses. Knowing these governance relationships gives us some vague idea about possible thematic roles, such as the Agent, the Action,

and the Patient. However, as Levin describes, different syntactic alternations may describe a common underlying meaning being conveyed; thus, only considering syntactic structure may not be enough to correctly identify the thematic roles. Another problem of syntactic parsing is that no semantic categorization of syntactic entities such as noun phrases, verbs, and adjectives are given. This is a particularly big problem for our problem domain, where recognition of the semantic categories is important for several tasks, such as semantic role extraction, semantic type deduction, and learning personal commonsense. While the understanding is not as deep, syntactic parsing is a good choice for the class of problems, which need a robust mechanism, but not as deep of an understanding. Syntactic parsing is robust because a lexicon of words indicating their parts of speech, as well as a context free grammar describing how lexical items combine into larger and larger constituents is sufficient for parsing. Though further language resources such as verb argument structure may be able to improve parses, they are generally not required for syntactic parsing to work *well enough*.

Semantic parsing lies at the opposite spectrum of the depth-to-robustness tradeoff. Unlike syntactic parsers, semantic parsers can produce representations that arguably give much more understanding. As we saw in Chapter 1, the Cyc-NL semantic parser uses frames to give highly specific semantic roles to concepts in the parse. UNITRAN parsed into an ontology of event primitives based on Jackendoff's Lexical Conceptual Structures. In UNITRAN, the representation of the meaning of a verb may require a very complicated structure, because in this system, all understanding must be reduced to the primitives. UNITRAN and Cyc-NL exemplify the deep understanding and brittleness commonly associated with semantic parsers. Cyc-NL requires an exceedingly large

number of situation frames to be able to parse a broad range of sentences with good coverage. UNITRAN also requires a tedious lexicon mapping words into LCS primitives. In general, semantic parsers require comprehensive and sophisticated language resources, and because such resources are hardly ever complete, semantic parsers are hardly ever robust over a large sampling of texts. Given its problems with robustness and broadness of coverage, at this time, semantic parsers can only be useful when used in controlled or well-constrained domains that require deep understanding.

While syntactic parsing solves the class of problems that require only very shallow understanding with relatively good robustness, and deep semantic parsing solves a class of problems that require deep understanding or reasoning, but can tolerate brittleness, we can identify a third class of problems that require a level of semantic understanding lying somewhere between what syntactic parsing and deep semantic parsing offer, and a level of robustness comparable to that of syntactic parsing. To utilize the robustness of syntactic parsing, such an approach might be built on top of a syntactic parse tree. To achieve a slightly deeper level of semantic understanding, semantic knowledge might be used to reformulate and transform the syntactic parse tree.

We argue that observational learning of annotations from email text belongs to this third class of problems. Since annotations include real world structures and concepts, it is important to be able to identify such things in the text, and to accomplish related understanding tasks such as learning personal commonsense, and it is helpful to have event structure information. It is also important to have robustness. In the Cyc-NL approach, a large set of semantic frames and primitive concepts would need to be defined. However, real world concepts that we target constitute too large a set to hard

code. In UNITRAN, a complex lexicon would be required, as well as verb-argument structure, among other lexical knowledge, but such knowledge is scarce and incomplete. So in our approach, robustness is rooted in the robustness of syntactic processing.

Our solution combines syntactic analysis, semantic typing, and event structure to achieve a moderate level of world semantic understanding, and a level of robustness and breadth of coverage comparable with syntactic analysis.

Given the sentence "Last weekend, I went to Ken and Mary's wedding in San Francisco," our semantic understanding mechanism will return the following event structure representation, which represents a world semantic reformulation of a syntactic parse tree.

```
[ASSERTION
 [PERSON I]
 [ACTION go
  [PATH to
    [EVENT
      [EVENT [PEOPLE [PERSON Ken] and [PERSON Mary/NNP 's]] wedding]
      [LOCATION in
        [PLACE San Francisco]]]]]]
```

The knowledge representation and ontological structures used in the above parse are discussed in more detail later in this chapter. In addition to this world semantic parse tree, two other mechanisms are needed for the automated learning of annotations. One is the identification of relevant sentences from a large body of email text. The other is the extraction of annotations given parsed relevant sentences.

This chapter is organized as follows. First, we present the automated annotation mechanism in greater detail. Second, we discuss how understanding agents are the crux of the semantic reformulation task. Third, we present the benefits of using a LCS event structure ontology. Fourth, an implementation of the semantic understanding machinery

is presented. We conclude the chapter with a discussion of some of the limitations associated with the approach given in this chapter.

2.1 Identification, Parsing, and Extraction

The automated annotation mechanism can be broken down into: the identification of sentences relevant to a photo, given a large body of text; the parsing of said relevant sentences using the world-semantic parser given in the overview of this chapter; and the extraction of important semantic roles from the relevant parsed text to serve as "annotations" for the photo. When working together, these three pieces constitute a mechanism that can process an email embedded with photos, and annotate photos with relevant descriptions from the text. Each of the three annotation mechanisms is now presented in greater detail.

Identification

Given a large body of text embedded with photos, how can the sentences relevant to the photo be identified? To be able to perform perfect identification of relevant sentences would require complete natural language understanding, and may require some commonsense reasoning, as well as some image analysis. Given that we are far from having that capability, we use a scoring approach to combine many different types of evidence of relevance, which we refer to as *relevance indicators*. We combine this scoring approach with a *context attachment strategy*.

Relevance indicators offer very diverse pieces of evidence for why a sentence might be relevant to a photo. The leading indicator is proximity to the photo. From empirical evidence with user tests conducted on an earlier version of ARIA at Kodak Labs

(Lieberman et al., 2001), it is clear that users generally describe the picture right before or sometimes right after where the picture is located in the text. Punctuation is also a remarkable indicator, because a sentence immediately preceding a photo, and ending in ":" is likely to be relevant. Cue words such as "here is a picture of" and "this is", and "posing" also offer evidence of relevance. While each relevance indicator offers a piece of partial evidence, it is the successful combination of these pieces of partial evidence that cause a sentence to be deemed relevant to a photo.

Once sentences whose scores meet a threshold are successfully added to the relevance context of a photo, a context attachment strategy is used to attach more sentences to the relevance context. In this strategy, given one sentence that was deemed relevant, other sentences can be pulled in, or attached to, the relevance context of the photo. The main feature used in context attachment is anaphor-to-referent placement. An anaphor is a phrase or pronoun that refers back to something previously said, called the referent. The referent is usually a noun phrase, though it can also be a whole clause, a whole sentence, or a whole paragraph. The idea is that if a sentence already deemed relevant has anaphors, and those anaphors point back into previous sentences, then we can include those sentences into the relevance context as well.

Combining these two approaches, the semantic understanding system described here can do an acceptable job of identifying relevant sentences, though admittedly imperfect because looking at features only approximates true understanding. For example, given the two input sentences: "Last weekend, I went to Ken and Mary's wedding in San Francisco. Here is a picture of the two of them at the gazebo," the identification process is able to successfully identify both sentences as containing relevant annotations. First,

the second sentence is identified as relevant because of its proximity to the photo, and because it contains the salient cue words "here is a picture of." Next, the context attachment process identifies the anaphor, "the two of them" as referring to two people. Anaphora resolution identifies the referent of "the two of them" to be "Ken and Mary" from the previous sentence, so that sentence is also attached to the relevance context of the photo.

Parsing

Once relevant sentences are identified, the world-semantic parser described earlier parses them into a Jackendoff-style event structure parse tree, with world semantic concepts identified and categorized. Because we rely on the robustness and broadness of coverage of a syntactic parser, the processing begins with a syntactic constituent parser of English. In addition, tagging parts of speech is done to get extra information on lexical items. The second phase of processing is reformulation. In this step, world semantic knowledge of concepts and their categories is infused into the syntactic parse tree. Understanding agents (UA), backed by knowledge bases of places, people's names, events, emotions, and regular expressions transform the syntactic structure tags into world semantic tags such as TIME, DATE, PERSON, PEOPLE, PLACE, THING, EVENT, EMOTION, ACTION, STATE.

After world semantic concepts are identified, a rule-based transformational grammar propagates these syntactic-to-semantic changes up from the leaves of the parse tree to the root node. For instance, one rule might appear as follows:

```
IF
     currentNode.label is "PERSON" and
     currentNode.parentNode.label is "PREP-PHRASE" and
     currentNode.parentNode.firstChild.value is "with"
THEN
     Set currentNode.parentNode.label to be "COAGENT"
```

When understanding agents are applied to all the terminal concepts in the syntactic parse "(PP with (NP Mary))", a tag change occurs and the following results: "(PP with (PERSON Mary))." Upon the application of the grammar rule above, the partial parse tree "(PP with (PERSON Mary))" becomes "(COAGENT with (PERSON Mary))." This is what propagating tag changes up the tree means.

One special type of grammar rule performs semantic importance deductions. For example, in the sentence, "Here is a picture of the bride and groom doing the hora," where "hora" is a traditional Jewish dance, "hora" will be learned by ARIA as an annotation, even though it is not a known lexical item and has a generic semantic type of THING. A semantic importance deduction rule makes possible the recognition that "hora" is an important thing. Since "the bride and groom" are semantically typed PEOPLE, which are defined to be good candidates for annotation, and since "the bride and groom" are "doing" (ACTION) the "hora," the semantic type of "hora" is promoted from THING to the type: THING_IN_PHOTO, which is a way to mark it for inclusion as an annotation. We examine semantic importance deductions more closely later in this chapter.

After all transformational grammar rules have fired, what results is a semantic tree with world-semantic concept tags (e.g. person, people, place, event, thing, emotion), Jackendoff-style event structure tags (e.g. state, action, property, purpose, manner, amount, time), and some thematic role tags (e.g. agent, coagent, patient, trajectory,

instrument). The specific ontology of tags used here were chosen because they were appropriate to our task. At the end of this stage, a semantic parse tree is produced. For the purpose of learning personal commonsense, this parse tree undergoes further manipulation and distilling, which we will talk about in a later chapter; but because the focus of this chapter is automated annotation, we move on to describe the third and final step in this mechanism, which is semantic role extraction.

Extraction

Before we can extract annotations, we need to agree on what sorts of annotations should be extracted from the text, and establish standards for annotation. This is an important question of knowledge representation. Before we proceed, we briefly describe and justify our knowledge representation for annotations.

The state-of-the-art systems for the annotation of images generally organize annotations into metadata fields. For example, in Allan Kuchinsy et al.'s Fotofile system (1999), structured metadata attributes provides a useful and uniform way to represent an annotation through the attributes of creation date, location, subject, people, title, and description. Such an organization has the benefit of being easy to understand from a user's point of view, and can also lead into interesting retrieval methods such as geographical visualization, etc. Our system uses a similar metadata organization to Fotofile, which we will refer to as a semantic annotation frame. The frame has four slots: Who, What, When, and Where. "When" can be obtained for free from the digital photo timestamp. "Who" encompasses people contained in the picture, "What" are things in the picture other than people, and "Where" is an event or location where the picture was

taken. We also refer to the "Who, What, When, and Where" as semantic roles played by the text.

Given a semantic parse tree for each relevant sentence from the previous processing step, concepts that have important semantic types are extracted and used to populate the annotation frame. By important semantic types, we mean concepts typed as PERSON and PEOPLE would fall under the "Who" slot of the annotation frame, concepts typed as THING_IN_PHOTO would fall under the "What" slot, and so on. In the case of multiple relevant sentences, annotations are merged additively, discounting duplicates, into a single annotation frame, which gets associated with its photo.

Through the three processing steps of identification, parsing, and extraction, descriptions from a large body of text can be used to annotate a photo with relevant annotations. The crux of this text understanding and annotation mechanism is in the transformation of syntactic constituents into semantic constituents. World-semantic understanding agents make this transformation possible. The next section elaborates on the mechanisms of these understanding agents.

2.2 World-semantic Understanding Agents

World-semantic understanding agents (UAs) help to add semantic expertise to syntactic parsing in our semantic understanding system. We say that these UAs are world-semantic because they pertain to the semantics of common everyday world concepts. Though they themselves do not constitute a semantic understanding of the system, UAs do help begin "digesting" the semantics of concepts, by recognizing the world semantic category that a concept belongs to, or by interpreting temporal phrases as

a date interval that computers can more easily understand, and so on. Each agent understands a different type of concept. Agents may employ whatever method or technique that best suits that type of concept, such as pattern-matching, and dictionary lookup. No agent by itself is intelligent, but by combining many agents, a semantic understanding agency can be constructed, to which we might more appropriately attribute intelligence. So what understanding agents might be needed to aid in the understanding of email text for the purpose of automated annotation? Consider the following:

"Hi Jane! Last weekend, I went to Ken and Mary's wedding in San Francisco. Here's a picture of the two of them at the gazebo."

To recognize the concepts in the above excerpt, several understanding agents may be useful. 1) A name UA, having a dictionary of male and female names and roles (e.g. "my mother"), may use pattern-matching to recognize people's names, by first and last name and gender. 2) A temporal UA having knowledge of different temporal phrases and ways to model time can be used to understand the phrase, "last weekend" as a date interval. 3) A place UA having knowledge of geographical places (i.e. cities, countries) as well as everyday places (e.g. park, school, café) can help to semantically tag concepts such as "San Francisco" and "gazebo" as being places. 4) An events UA, having knowledge of everyday events, should understand that "wedding" is an event based on its dictionary knowledge of events, and based on syntactic constraints on events. 5) An anaphora UA should understand that "the two of them" is an anaphor referring to "Ken and Mary."

The understanding agents enumerated here use a few common strategies. The most common strategy is having dictionary knowledge of the membership of a concept within some semantic category. Pattern-matching is another approach, as used with names,

time, and places. Constraint matching is a third approach, as used in the anaphora resolution UA, and events UA. The important lesson learned is that understanding agents themselves cannot be perceived to be intelligent because each one "understands" very little. In fact, they may even be quite naïve. However, they make possible deeper understanding in our mechanism, namely, understanding the event structure of a story text. This is the focus of the next section.

2.3 Event Structure

While world-semantic understanding agents help to predigest the semantics of a storytelling text, they only enable very shallow understanding, comparable to keyword spotting techniques. However, when combined with syntactic constituent structure and rule-based knowledge tying constituent structure to event structure, a deeper understanding can be achieved.

Event structure information poured over a foundation of syntactic constituent structure cannot claim to achieve the level of understanding found in UNITRAN or the level of understanding intended by Jackendoff's Lexical Conceptual Structures (LCS). In the case of UNITRAN, semantics is compositional and reminiscent of Montague. Each lexical item is reformulated as a function of a small set of LCS primitives, as shown below.

```
[GO_Ident
  ([Thing I],
   [TOWARD_Ident
        ([AT_Ident
            ([Thing I],
             [Location KEN AND MARY'S WEDDING])])])]
```

The advantage of such compositional semantics is that the resulting representation is not tied to syntactic structures or preferring specific surface realizations. This makes the UNITRAN approach especially appropriate to interlinguas. In contrast, our use of event structure is not only dependent on specific surface realizations, but rather, event structure information is built on top of syntactic structures, as shown below.

```
[ASSERTION
 [PERSON I]
 [ACTION go
  [PATH to
    [EVENT
      [EVENT [PEOPLE [PERSON Ken] and [PERSON Mary/NNP 's]] wedding]
      [LOCATION in
        [PLACE San Francisco]]]]]]
```

The justification for doing so is two-fold. First, we can rely on the broad-coverage nature and relative robustness of syntactic parsing, because the task of automated annotation calls for rather robust processing technique. Second, event structure, even when built on syntactic structure, is still useful in performing not-so-deep understanding tasks not achievable without event structure, such as simple semantic type deduction, and user modeling, which are the subjects of the next section.

UNITRAN uses an ontology of primitives based on the ontology suggested by LCS. These primitives are organized into conceptual categories. Taken from Dorr, 1992, the table in Figure 5 shows a subset of the conceptual categories and primitives used in UNITRAN.

Category	**Primitives**
EVENT	CAUSE LET GO STAY
STATE	BE GO EXT ORIENT
POSITION	AT IN ON
PATH	TO FROM TOWARD AWAY FROM VIA

THING	BOOK PERSON REFERENT KNIFE-WOUND, KNIFE, SHARP-OBJECT, WOUND, FOOT, CURRENCY, PAINT, FLUID, ROOM, SURFACE, WALL, HOUSE, BALL, DOLL, MEETING, FROG
PROPERTY	TIRED HUNGRY PLEASED BROKEN ASLEEP DEAD STRETCHED HAPPY RED HOT FAR BIG EASY CERTAIN
LOCATION	HERE THERE LEFT RIGHT UP DOWN
TIME	TODAY SATURDAY
MANNER	FORCEFULLY LIKINGLY WELL QUICKLY DANCINGLY SEEMINGLY HAPPILY LOVINGLY PLEASINGLY GIFTINGLY UPWARD DOWNWARD WITHIN HABITUALLY

Figure 5: A Subset of Conceptual Categories and Primitives used in UNITRAN (Source: Dorr, 1992)

The LCS language is actually made up of many different elements beyond just the conceptual category (also called conceptual parts-of-speech) and primitive shown above. The main elements are: conceptual categories, semantic fields, and primitives. Minor elements are conceptual variables, semantic features (e.g. eatable, drinkable), constants i.e. non-decomposable concepts (e.g. money), and lexical functions.

In contrast to the large ontological count prescribed by LCS, our mechanism has a smaller ontological unit count. Ontological items as we define them are less granular than primitives described by LCS; they more or less lie at the same level of granularity as the conceptual categories of LCS, which are: *thing, event, state, place, path, property, purpose, manner, amount, time*. In addition to LCS primitives and types, our event ontology also includes some thematic role categories such as *agent,* and *patient.* Figure 6 shows some of the primitives used by our mechanism.

Partial list of primitives
EVENT, PERSON, PEOPLE, PLACE, TIME, MANNER, STATE ("to be"), ACTION, PATH, ORIGIN, DESTINATION, THING, EMOTION, PROPERTY, LOCATION, TRAJECTORY, AGENT, PATIENT, CO-AGENT, INSTRUMENT

Figure 6: Some of the primitives used in ARIA

Unlike the ontology used by UNITRAN, which requires an extensive lexicon to ground the semantics of words into the UNITRAN primitives, the ontology used by ARIA can be applied directly to the syntactic constituents, after understanding agents have labeled the text. This ontology is very lightweight, and one limitation is that understanding is still somewhat tied to surface realization factors, such as word choice. For example, let's reconsider the sentence, "last weekend, I went to Ken and Mary's wedding in San Francisco." In UNITRAN, the sense of the word "go" versus the other possible word choice of "attend," may end up mapping into the same primitives, which is the often sought after "unification" problem in computational linguistics. However, using ARIA's ontology, it is not possible to completely unify the two words "go" and "attend."

Though the event ontology in ARIA greatly abridges the set of primitives in LCS, we believe the spirit of LCS is preserved. Our ontology is still organized about the notion of motion, and of change. For example, change of physical location, change of possession, change of property and so on. Detecting change is vitally important to tasks ARIA will need to do, such as user modeling. By detecting the change of the physical location of the user in the story text, it is possible to record all the places the user has visited. By

detecting emotional change of the user, it is possible to isolate how the user feels about a subject.

We apply the event ontology to transform syntactic constituent structure into event structure by mapping syntactic constituents tags into our event structure. Transformation takes place through the iterative application of rules from a transformational grammar. This begs the question: How can we expect the supposedly semantic-natured event structures to share the same concept constituent structure as the syntactic parse? The best answer we can offer is that syntactic and semantic structures are likely to be similar because syntactic similarity is suggestive of semantic similarity, though this assertion does not always hold. Levin's verb classes and alternations, for example, depend on the semantic to syntax similarity condition. Another reason to believe that event structure can be built over syntactic constituents is that an important characteristic of LCS is that it is closely related to syntax, much like a kind of X-bar semantics (Jackendoff, 1993). Therefore, our event ontology is likely to fit well over syntactic structures such as prepositional phrases (PP), noun phrases (NP), and verb phrases (VP).

Using event structure built on top of syntactic structure, it is possible to perform two understanding-enabled tasks that are important to ARIA. Because these tasks exploit knowledge about event structure, we refer to them more specifically as structural deductions.

2.3.1 Structural Deductions

Thus far we have presented a world semantic parsing mechanism built on top of syntactic constituent structure. We have argued that the proposed mechanism provides a

proper level of semantic understanding needed to accomplish ARIA's goals of automated annotation and learning personal commonsense (user modeling). At this point, one may question why all this natural language processing is even necessary. Let us consider the following criticism.

The pragmatist asks:

> *Why go through all this effort to parse text? Why not just keyword spot PLACES, EVENTS, NAMES to include as annotations?*

Responding to the criticism, we first point out some of the shortcomings of the keyword spotting approach which makes it unsuitable for ARIA's tasks. Admittedly, keyword spotting is a rather robust approach, is easy to implement, and shows good separation of knowledge and mechanism.

The most naïve keyword spotting has very little knowledge. One can envision a keyword spotter that picks out any keyword not appearing in some "stop list" of words (also known as exclusion lists) to associate with a photo as an annotation. The stop list would consist of words that are commonly thought of as being somewhat semantically empty, such as determiners, prepositions, modals, pronouns, and so on. This method is a bit too naïve because it spots keywords rather than concepts. For example, performing keyword spotting over the sentence, "Last weekend, I went to Ken and Mary's wedding in San Francisco," would likely return the annotation array, ["last", "weekend", "I", "went", "Ken", "Mary's", "wedding", "San", "Francisco"]. Notice that in the array, concepts such as "last weekend," and "San Francisco" have been separated. This is not desirable because breaking up conceptual atoms changes its semantics. For example, breaking up "Golden Gate Bridge" into individual words would cause a photo annotated with the three words to incorrectly match text which bears very little semantic relation to

the intended concept, such as "golden grahams," "Harvard Gate," and "Brooklyn Bridge."

The pragmatist might suggest a slightly more informed keyword spotter would spots concepts rather than single words. To be able to spot a concept, the mechanism would need a dictionary of concepts that it wanted to spot. Such a dictionary could even be broken up into broad semantic categories such as PEOPLE, THING, PLACE, and EVENT. We would like to point out the limitations of this approach.

First, just because a concept is spotted in a text does not make it relevant to a photo. Keyword spotting has recognition knowledge, similar to our understanding agents, however, without structural information, it would not be possible to determine if a spotted concept bears semantic relevance to a photo. To naïvely include all spotted concepts as annotations would lead to many noisy, irrelevant annotations.

Second, keyword spotting cannot anticipate all concepts that may be relevant to a photo, because its lexicon cannot contain all possible concepts. For example concepts, which fall under the category, THING, cannot all be enumerated. For example, in the sentence, "Here is a picture of John at the Foo Bar Museum," the concept "Foo Bar Museum" is clearly a relevant annotation, but a pure keyword spotter will not be able to pick up on this because it is absent from the lexicon. By contrast, our semantic parse contains structural information, which can not only identify "Foo Bar Museum" as a concept (syntactic constituent recognition), but can also guess that it is a PLACE (deducing semantic type), and that it is relevant to the photo based on the structure of the sentence (deducing semantic importance).

Third, many types of concepts are variable in nature and cannot be recognized by a static lexicon. For example, the phrase "last weekend" is an important concept, but one which changes itself depending on the current temporal reference point. One week later, the same concept would be "two weekends ago," etc. Names are another example. It is possible to refer to someone by a first name, or Mr. Last Name, or full name, etc. To "spot" such concepts, some pattern-matching is necessary, and so a mechanism similar to our understanding agents would be needed.

Fourth, keyword spotting will not help ARIA do user modeling. Suppose we wanted to add to the user model all the places a user has traveled to. So for the sentence, "Last weekend, I went to Ken and Mary's wedding in San Francisco," ARIA might learn that the user has traveled to San Francisco, and more precisely, that the trip took place in the date interval corresponding to the phrase "last weekend." Without knowing the event structure relationship that the user physically moved his location to San Francisco last weekend, a mechanism cannot confidently learn this piece of knowledge in the user model. Keyword spotting does not give us any tools for this.

Revisiting the pragmatist's criticism, keyword spotting would make a great baseline approach to measure the success of our semantic parsing approach, but it is not powerful enough to suit our task because of its important limitations in not being able to pick up unknown concepts or variable concepts (e.g. names, temporal expressions), not being able to deduce semantic type and semantic importance, and not being able to assist in user modeling.

Let us now describe some of the tasks that our semantic parsing approach can perform, given event structure and semantic typing capabilities. They are: recognizing

unknown concepts and deducing their semantic type; deducing the semantic importance of a concept to a photo; and understanding event structure relations for user modeling.

Semantic Type Deduction

Given event structure, it is possible to deduce the semantic type of an unknown concept. For example, consider the sentence, "I went to a wedding in FooBar Sisco," the concept "FooBar Sisco" will not be recognized immediately as a PLACE, but given the structural rule that a person can go to an event, which may have some location, GO($person, $event, $place), the mechanism can guess, with some acceptable confidence score, that the semantic type of "FooBar Sisco" is PLACE.

Deducing Semantic Importance

Just because a concept falls under a semantic category that may be relevant to a photo does not mean that it *is* relevant. The most common examples of this are concepts in the semantic category THING. For example, consider the sentence, "Here is a headshot of Mark." Even though "headshot" is recognized as a THING, it clearly is not the subject of what is depicted in the photo. Instead, "Mark" is the subject of what is depicted in the photo. To help arbitrate the relevance of a THING to a photo, we need to utilize structure to deduce the semantic importance of the concept in question. A subset of the transformational grammar rules, which transform the syntactic parse tree to event structure, performs the task of deducing semantic importance. Not all THINGs are created equal. For example, in the example given earlier, "Here is a picture of the bride and groom doing the hora," where "hora" is a traditional Jewish dance, "hora" will be learned by ARIA as an annotation, even though it is not a known lexical item and it has a

generic semantic type of THING. A semantic importance deduction rule makes possible the recognition that "hora" is an important thing. Since "the bride and groom" are semantically typed PEOPLE, which are defined to be good candidates for annotation, and since "the bride and groom" are "doing" (ACTION) the "hora," the semantic type of "hora" is promoted from THING to the type: THING_IN_PHOTO, which is a way to mark it for inclusion as an annotation.

Event Structure for User Modeling

Event structure provides a concise representation for determining change. This is a great tool for user modeling. Without delving too much into the details of using event structure for user modeling, which is covered in Chapter 4, we give two examples of aspects of the user model that can be learned with event structure. First, consider the sentence "Last weekend, I went Ken and Mary's wedding in San Francisco." Using event structure information, we know that the user changed his/her physical location to San Francisco, and that it happened last weekend. This can be added to the user model as a piece of personal commonsense as follows: "Someplace the user has visited is San Francisco." Alternatively, other pieces of personal commonsense about the same sentence can be learned, such as, "The event, Ken and Mary's wedding, took place in San Francisco," or "The user was in San Francisco between 5/1/2001 and 5/3/2001."

In another example, the sentence, "In my office, Mark is always the funniest character," tells us about the relationship between Mark and the user's office. Event structure tells us that Mark's physical location (insofar as this sentence claims) is the user's office. Though seeing one occurrence of this relationship may not be important, multiple mentions of Mark being "in, at, around" the user's office can lead ARIA to

conclude the fact, "The person Mark is often found in the user's office." This piece of knowledge can be helpful in improving retrieval. Every time the user's office comes up in the text, we can pull up photos annotated with Mark.

Having presented the three steps in our automated annotation task i.e. identification, parsing, and extraction, and having defined and discussed event structure and its applications in our semantic understanding task, we now present WALI, the semantic understanding subsystem implemented in ARIA which is responsible for automated annotation, and also facilitates the personal commonsense learner and photo retrieval mechanisms.

2.4 WALI: World-Aware Language Interpreter

The *World-Aware Language Interpreter* (WALI) is the ARIA semantic understanding subsystem which implements the semantic understanding mechanism discussed in this chapter thus far. As suggested by its namesake, the emphasis and the innovation of WALI is that it is world-aware. WALI recognizes concepts from the everyday world, but not just those that can be found in dictionaries, such as "playground," and "house." The semantic knowledge behind WALI does not come from a dictionary or encyclopedia, but rather, is mined out of a large-scale corpus of commonsense knowledge about the everyday world.

For the automated annotation mechanism, WALI, when passed a body of text, can identify sentences relevant to a photo, can produce an world semantic event structure parse of the text, and can extract semantic roles of Who, What, When, and Where from

the text. Additionally WALI also provides event structure understanding services to the Personal Commonsense Learner mechanism, and semantic role extraction and temporal phrase resolution services to the retrieval mechanism.

In this section, we present the implementation of WALI and describe the performance, capabilities, and limitations of this implementation.

2.4.1 An NLP Architecture

WALI's natural language processing architecture can be thought of as having three phases: processing text, syntactic analysis, and semantic analysis. Each phase can be further decomposed into smaller stages. As shown in Figure 7, WALI has eight processing stages.

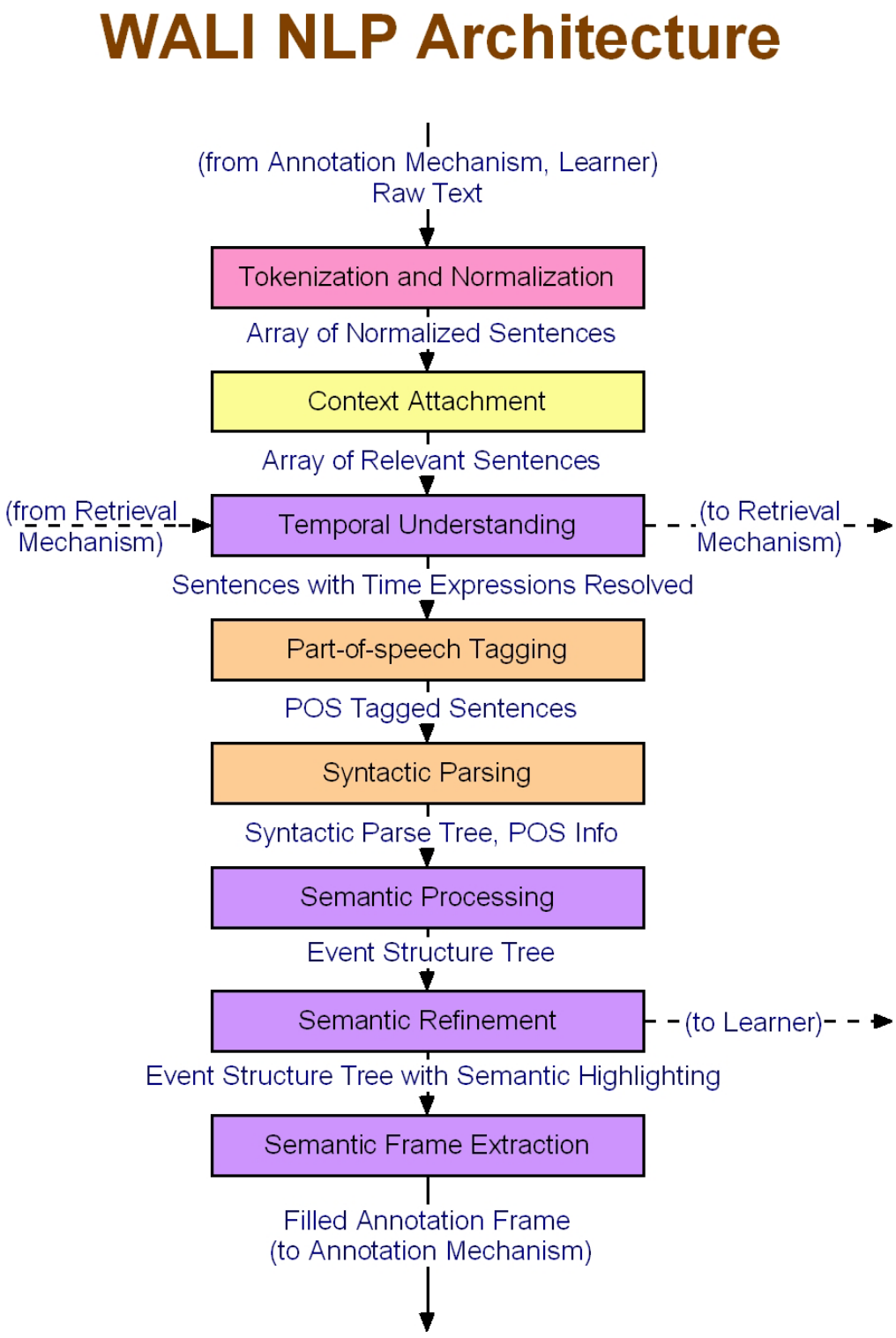

Figure 7: Processing steps in the WALI Architecture.

We give a brief overview of each stage below, followed by a more detailed review of the eight stages in the ensuing sections.

1. **Tokenization and Normalization.** Raw text inputted from ARIA needs to undergo a basic normalization routine before being sent to parsing, including space and contraction normalization, and sentence boundary detection.
2. **Context Attachment.** Normalized and separated sentences are judged for their possible relevance to a photo based on anaphor-to-referent attachment, and a variety of structural features. Relevant sentences get "attached" to the relevance context space of a photo. Each relevant sentence undergoes the remainder of the processing steps individually, until the last stage of semantic frame extraction, where the results are merged.
3. **Temporal Understanding.** A normalized sentence runs through temporal pattern matchers to map a variety of expressions about dates and date intervals into concrete date intervals understandable by a computer.
4. **Part-of-speech tagging.** Part-of-speech information is needed for mapping syntactic structure into world-semantic event structure. Since POS tags are not available through the syntactic parser used in WALI, a separate POS tagging effort is made here.
5. **Syntactic Parsing.** Syntactic parsing takes in a normalized sentence and returns a government and binding constituent structure (c-structure), in the form of a parse tree, whose leaf nodes are words, and intermediary nodes are syntactic constituents.

6. **Semantic Processing.** Taking in a syntactic c-structure and POS information, the semantic processing stage runs understanding agents on the c-structure, and then a transformational grammar runs and reformulates the syntactic constituent tags into event structure tags.
7. **Semantic Refinement.** Performing structural deductions on the parse tree further refines the event structure output from the previous processing stage. Guesses are made to assign semantic types to unknown concepts. Concepts in important semantic roles are semantically highlighted.
8. **Semantic Frame Extraction.** Semantically highlighted event structures are passed as input to a heuristic extraction mechanism in this stage. Relevant concepts are extracted into a frame whose slots are WHO, WHAT, WHEN, and WHERE. If there were multiple sentences being processed, the semantic frames of all of them would be combined additively into a new frame.

Each section below provides a more detailed view of the eight stages in the WALI architecture.

2.4.2 Tokenization and Normalization

Raw text is fed in from the ARIA annotation mechanism and personal commonsense learner mechanism. Included with the text are placeholders for the positions of embedded photos. The processing phase prepares text for the pattern-matching and parsing mechanisms of later stages, which all rely on the text having normalized spacing, punctuation, and clearly drawn out sentence boundaries. This module normalizes spacing, contractions and slang, and separates text into sentences.

The spacing convention implemented separates all punctuation from words except in the cases of abbreviations, where punctuation stays contiguous to the letters. Abbreviations are detected using a naïve dictionary lookup into a 300 list of common abbreviations. Possessive ending " 's " is also separated away from the word it modifies, as required by most syntactic parsers.

Roughly 40 common contractions like "ain't," "aren't," and "didn't" are expanded out. Email slang such as "dunno," "gimme," "gonna," and "u" are also expanded into their conventional equivalents.

Punctuation marks, amount of spacing after punctuation, and capitalization after spacing are some of the cues used in sentence boundary detection. Checking against an abbreviations dictionary is also necessary to disambiguate an end-of-sentence period from an end-of-abbreviation period. Currently, discourse structure and language models are not used for sentence boundary detection.

Once text is normalized, sentences are separated and put into an array. This sentence array is sent on to the context attachment module to determine the relevance of each sentence to a particular photo.

2.4.3 Context Attachment

Given an array of sentences and a pointer into the text where a photo is placed, the context attachment module attempts to return a subset of the sentences that can provide relevant annotations for the photo. The algorithm calls for two phases. In the first phase, all sentences are evaluated with a relevance score, based on a diverse set of heuristic features that we call relevance indicators (RI). Sentences scoring beyond a threshold are

pulled into the relevance context space. In the second phase, each sentence in the relevance context space that contains anaphora can pull in other sentences into the relevance context, which contain the referent for each anaphor.

Relevance indicators evaluate each sentence with a relevance score from –3 to +3. Negative scores suggest irrelevance, positive scores suggest relevance, and a zero score suggests no conclusion. Scores from all relevance indicators are heuristically weighted by the "salience" of each RI, and the weighted scores are summed for each sentence. All sentences scoring above a threshold are put into the relevance context space of the photo.

The relevance indicators implemented are:

1) Proximity to the photo (the closest sentence is preferred)
2) Cue words indicating a caption or photo description, such as "here is a picture of," "this is," and "posing."
3) Punctuation marks, e.g., a sentence preceding the photo, ending in ":" or "…" is more likely to be relevant.
4) Paragraph breaks indicating a change of topic. Generally, sentences one or more paragraphs away from the photo will score low in the RI.

Once sentences are added to the relevance context space of a photo, each sentence is searched for anaphora, and more specifically, common pronouns like "him," "her," "them," "they," "there," "then," "it," etc. After identification, a knowledge poor algorithm similar to that described by Mitkov (1998), is used to identify a referent for each anaphor, usually from immediately preceding clauses and sentences. More specifically, the system looks for noun phrase referents, as opposed to other possible types of referents such as whole clause, whole topic, or existential (i.e. "it is raining

outside"). Referents are verified by ensuring number (plural/singular/counting), gender (male/female), and semantic type (person/place/time/thing) agreement between anaphor and referent. All sentences containing referents are also added to the context space.

2.4.4 Temporal Understanding

The temporal understanding agent maps temporal expressions into date and time intervals that are intelligible to the computer. Regular expression pattern-matching is employed to efficiently recognize temporal expressions. Four types of temporal expressions are handled in the system.

1) **Calendar date and time expressions.** This class of expressions makes use of calendar expressions such as months and seasons. Example of these phrases include: "1/1/2001," "January 1st," "last March," and "summer of 1999." Generally, computing dates and date intervals in this class of expressions requires some knowledge of the calendar, i.e. number of days in the month.
2) **Common date and time expressions.** Common time expressions generally do not require calendar knowledge. Examples are highly common expressions such as "two weeks ago," "last weekend," "Monday." Most of these expressions use the current date and time as the temporal reference point.
3) **Holiday expressions.** Holiday expressions require knowledge of the occurrences of named holidays. For example, "Christmas, 2001," "last Halloween," and "labor day weekend" are examples are holiday expressions. While some holidays have fixed dates, others vary from year to year, such as

Mother's Day, Labor Day, etc. Usually however, variable date holidays are easy to calculate. In our implementation, only major US holidays are encoded. The most difficult holiday to calculate is Easter. Not only is it defined to be on a different day for the Eastern Orthodox faith, but also a complicated formula is needed to make the calculation. Eastern orthodox Easter is not supported in our system.

4) **Referential date and time expressions.** This is the class of expressions that measures date and time with respect to a specific temporal reference point (other than the present date and time). Examples of these phrases include: "the day before last Monday," "the week before Christmas," etc. Note that both singular dates/times as well as date and time intervals are possible, and are handled. These phrases are generally handled as a post-modification to an existing recognized temporal phrase; therefore, we also refer to this class of expressions as being compound expressions.

After recognition, these temporal phrases are mapped into a computer representation of time, appearing in one of four formats:

ARIA_TIME{00h:00m}
ARIA_TIMESPAN{00h:00m-00h:00m}
ARIA_DATE{00m00d0000y}
ARIA_DATESPAN{00m00d0000y-00m00d0000y}

2.4.5 Part-of-speech Tagging

Part-of-speech (POS) information is needed for the semantic processing phase, but our syntactic parser does not use, or give us POS tags for words. The purpose of this module is to independently calculate POS information on words in the text, so that these

tags can be supplied to the semantic processing module. The POS tags chosen for this module come from the popular Penn Treebank tagset (Marcus, Santorini, and Marcinkiewicz, 1993). Brill's rule-based transformational-based learning approach to tagging (1995) is implemented. The original POS tagging rules file and lexicon, which accompanied his implementation, were incorporated into a Java version implemented specifically for this thesis.

2.4.6 Syntactic Parsing

From the three publicly available broad-coverage syntactic parsers for English, the Link Grammar parser was chosen for our implementation because it is robust, broad-coverage, and outputs constituent structure. Link grammar is an original theory of English syntax, which considers a set of labeled pairwise linkages between words. The parser has about 60,000 lexical items (including word form variation), and covers many different syntactic constructions including idiomatic language. Link grammar theory is rather robust, allowing for partial parsing by skipping over unintelligible portions of sentences. Link grammar can handle unknown vocabulary and make guesses as to a word's syntactic type. Link Grammar is able to output into constituent structure, even though the internal representation of the text consists of labeled linkages between words. No part-of-speech tag information is used or generated in the parsing process, because link grammar has its a unique theory of syntax.

Two fundamental limitations of link grammar are that its constituent structure output is not as robust as its linkage structure output, and the parser was trained on "clean" corpora, not dirty and noisy email corpora. Whereas a link grammar parse tree

represented through linkages is rather robust, having the ability to skip over portions of the sentence it cannot understand, a link grammar constituent output is not as robust. The problem is that pairwise linkages are rather like flat relations, where the flatness lends itself well to partial parsing capabilities; however, constituent structure is hierarchical, and therefore, less tolerant of partial failure. The second limitation is that link grammar relies heavily on knowledge of capitalization, numerical expressions, and punctuation. However, email texts are generally "mildly ill-formed," because they involve slang, and may not use proper capitalization, and punctuation. Therefore, in the absence of reliable lexical and punctuation cues, parser accuracy may decrease.

The constituent output of link grammar parser uses the standard constituent groupings that most government-and-binding grammar parsers use: S (sentence or clause), VP (verb phrase), NP (noun phrase), and PP (prepositional phrase), among others. The parser output is read into a parse tree data structure.

2.4.7 Semantic Processing

The transformation from syntactic constituent structure to event structure with world-semantic typed concepts begins with world-semantic understanding agents recognizing and recasting the semantic categories of concepts in the parse tree. In the second phase, part-of-speech information is combined with a transformational grammar to transform the parse tree into an event structure tree.

Understanding Agents

Understanding agents (UA) for PERSON, PEOPLE, PLACE, THING, EVENT, EMOTION, ACTION, and STATE are serially run on the parse tree. The understanding agent for dates and times ran earlier, as a separate module.

The understanding agents for PERSON and PEOPLE look for syntactic constituents of type NP (Noun Phrase) and perform pattern-matching for recognition. Having knowledge of 5000 common male and female first names, and regular expression models of different ways in which a name can be expressed (e.g. first name, first and last, Mr./Mrs./Ms. Last, etc), recognized names are labeled with PERSON. Personal pronouns and semantic roles played by people (e.g. father, sister, friend, etc) are also recognized. Capitalization and possessive pronouns like "my" and "our" are also cues for recognition. The understanding agent for people recognizes common groups of people (e.g. guests, protesters, etc.) and also two or more PERSON types connected by conjunction (i.e. "and").

The understanding agents for PLACE, THING, EVENT, EMOTION use keyword spotting. Approximately 2,000 concepts were mined out of the Open Mind Commonsense (OMCS) corpus. OMCS will be discussed in more detail in Chapter 3. For now, it suffices to say that the OMCS ontology of commonsense relations, taking the form of fill-in-the-blank sentences, provides sufficient semantic and syntactic constraints on the blanks to allow concepts to be mined out of those sentences, and to be able to give semantic category classification to those concepts. Concepts mined out of OMCS were organized into an ontology of semantic categories including the ontological items: PLACE, THING, EVENT, EMOTION. These large knowledge bases are used to provide semantic classification to constituents in the syntactic parse tree. NP (noun phrase), VP

(verb phrase) and AdjP (adjective phrase) constituents are examined. In the current implementation, there is no conflict resolution or disambiguation strategy to arbitrate the classification of a concept into two or more potential semantic categories. The classification of concepts is taken in a hierarchical order of the available semantic types. In the future, it may be desirable to associate uncertainty weights with the classification of concepts under each semantic category so that a disambiguation strategy could be used. Additionally, it may be desirable to contextualize the keyword spotting technique by looking at syntactic and semantic cues in the context of the concept that can aid in classification. Contextualization is not currently implemented because it would require too much custom knowledge engineering. Instead, the most scalable solution was implemented. One note on the recognition of concepts of the semantic category, THING: all unknown syntactic NP constituents will inherit the type THING, but it is up to the next processing module, Semantic Refinement, to determine the relevance and semantic importance of each THING to the photo.

ACTION and STATE are semantic categories of verbs. STATE verbs describe a static state, and the most prominent verb in this class is "to be." ACTION verbs encompass all other verbs not included in STATE. Currently, the function of ACTION and STATE is not to divide verbs into functional categories as with Levin's verb classes, but the function is to simply normalize verbs to their lexemes (base form) so that they are easier to process. Verbs are reduced to their lexeme by eliminating tense, and conjugations. This is done with a set of morphological rules, and an exceptions list of irregular verbs. In the future, it may be desirable to group verbs into functional categories, using a resource such as Levin's verb classes.

Transformational Grammar

After understanding agents have classified constituent concepts into semantic categories, a transformational grammar converts the remaining syntactic constituents into event structure constituents. Syntactic-to-semantic changes are propagated up the parse tree from the leaves to the root node. A set of example rules is given below.

```
IF
     currentNode.label is "PERSON" and
     currentNode.parentNode.label is "PREP-PHRASE" and
     currentNode.parentNode.firstChild.value is "for" and
     currentNode.parentNode.firstChild.pos is "IN"
THEN
     Set currentNode.parentNode.label to be "BENEFICIARY"


IF
     currentNode.pos is "DT" and
     currentNode.value is not in ["the"] and
     currentNode.parentNode.label is "NOUN-PHRASE" and
THEN
     Set currentNode.label to be "AMOUNT"


IF
     currentNode.label is "ARIA_TIME" and
     currentNode.parentNode.label is "PREP-PHRASE" and
     currentNode.parentNode.firstChild.value is "at" and
     currentNode.parentNode.firstChild.pos is "IN"
THEN
     Set currentNode.parentNode.label to be "TEMPORALPOINT"


IF
     currentNode.label is "PLACE" and
     currentNode.parentNode.label is "PREP-PHRASE" and
     currentNode.parentNode.firstChild.value is in THROUGH-synset
THEN
     Set currentNode.parentNode.label to be "PATH"
```

Components of rules can include the value of a node, the semantic/syntactic label of a node, the part-of-speech of a node, the parent node, and sister nodes. As illustrated by the last example rule, predefined synonym sets (synsets) are also components in rules.

Rules are fired in a most recently used order, and semantic processing is complete when no more rules fire. An event structure tree is returned.

2.4.8 Semantic Refinement

Semantic refinement entails the post-processing of the event structure tree to accomplish two goals. First, guesses are made to assign semantic types to unknown concepts, which are all marked "THING" from the previous processing stage. Second, semantic types in important event roles are semantically highlighted.

Guessing Semantic Types

In order to assign semantic types to unknown concepts, labeled "THING," syntactic and event role cues are used. Assigning semantic types to these concepts using only simple event role and syntactic cues amounts to guessing. But they are guesses made in the context of storytelling with photos, so these guesses are quite specific to ARIA. Some examples of guessing rules are as follows:

```
IF
     currentNode.status is "UNKNOWN" and
     currentSentence.getMainAction is in go_synset and
     currentSentence.getAgentNode.label is in ["PERSON","PEOPLE"] and
     currentSentence.getIndirectObjNode is currentNode.parentNode and
     currentNode.parentNode.firstChild.value is in LocationPrepSynset
THEN
     Set currentNode.label to be "PLACE"
     Set currentNode.guess_p to be "yes"

IF
     currentNode.status is "UNKNOWN" and
     currentSentence.getMainAction is in go_synset and
     currentSentence.getIndirectObjNode.label is "PATH"
THEN
     Set currentNode.label to be "PERSON"
     Set currentNode.guess_p to be "yes"
```

The rules presented above are merely informed guesses. Without more understanding and more knowledge, it is not possible to know for certain the correct semantic type. In explaining how to read these rules, we walk through the first rule. It states that if a concept node is unknown, and is in a prepositional phrase complement slot, where the preposition head is a location word such as “in”, ”near”, ”through”, ”at”, etc., and the main action of the event is “to go” and the main agent is a PERSON, then the unknown concept is probably a PLACE, because in the photo storytelling domain, people typically go to places.

Semantic Highlighting

Even though the semantic processing module has identified the semantic types of sentences, which have been deemed by the context attachment module to be relevant to a photo, not every known semantic type like PERSON, PLACE, and THING that is eligible to be an annotation *should* be an annotation. The purpose of semantic highlighting is to heuristically identify semantically typed concepts that are definitely relevant. Many of the same cues used to identify a sentence as relevant, in the context attachment module, are now used to semantically highlight concepts.

With regard to the PERSON type, in the sentence, “I took this picture of John and Mary,” it is clear that even though “I” was recognized as a PERSON, “I” is probably not a subject of the photo. In semantic highlighting, the semantic types of “John” and “Mary” are promoted from PERSON to PERSON_IN_PHOTO, whereas “I” is not.

Concepts with the type THING are in particular need of semantic highlighting because all unrecognized concepts are of type THING, yet clearly they should not all be included as annotations. For example, in the sentence, “I took this picture of the house,”

the concepts "picture" and "house" are both of type THING, yet "house" is definitely a subject of the photo and worthy of becoming an annotation, whereas "picture" is definitely not a subject of the photo. Semantic highlighting in this example would result in the type of "house" to be promoted to THING_IN_PHOTO.

The heuristics used for semantic highlighting are specific to ARIA's domain because they exploit cue patterns such as "picture of *x*," "here is *x* posing," etc. Semantic highlighting is implemented as regular expression pattern matchers, matching over the event structure tree.

A common question that might be asked is whether or not concepts that are not semantically highlighted will still be included as annotations. The answer is: Some will. Semantic highlighting merely brings definite annotations to the attention of the semantic extraction algorithm, but that algorithm will still include unhighlighted concepts, as appropriate.

2.4.9 Semantic Extraction

The semantic extraction algorithm takes as input the semantically highlighted event structure of each sentence in the relevance context space. The goal of this module is to extract relevant annotations from each sentence, and then merge annotations into an annotation frame, to be returned to ARIA's annotation mechanism.

The algorithm for extracting relevant semantic roles is as follows:

1. From each sentence, semantically highlighted concepts are first to be extracted. THING_IN_PHOTO fits in the "What" slot; PLACE_IN_PHOTO and EVENT_IN_PHOTO fit in the "Where" slot; PERSON_IN_PHOTO and

PEOPLE_IN_PHOTO fit in the "Who" slot; TIME_IN_PHOTO, TIMESPAN_IN_PHOTO, DATE_IN_PHOTO, and DATESPAN_IN_PHOTO fit in the "When" slot.

2. Semantically typed but not highlighted concepts which are a child node or sister node of a semantically highlighted node are included as annotations. The rationale behind this is that semantic type importance propagates across concepts in conjunction or disjunction, and propagates downward into nested structures.
3. In sentences where no annotation is semantically highlighted, all semantically typed objects other than THING (too generic, too noisy) are included in the annotation frame
4. The annotation frames generated by all relevant sentences are additively merged, minus duplicates, into a final annotation frame, which is returned as output.

2.5 Limitations

There are two types of limitations that deserve discussion. There are the limitations of the WALI implementation, and then there are the limitations of the approach to semantic understanding taken in this thesis. We discuss each type in a separate subsection below.

2.5.1 Limitations of the Implementation

As one might expect, the limitations of WALI amounts to the sum of the limitations of each of its modules. By far, the biggest performance bottleneck in the system is the syntactic parser. There are also some limitations presented by the accuracy of the part-of-speech tagger, knowledge limitations on the understanding agents, and limits on recognizable semantic types.

Syntactic parsing should be relatively reliable and robust. The Link Grammar Parser of English, used in this implementation, is actually rather robust, capable of skipping unknown words and phrases but not breaking the entire parse. Unfortunately, its robustness in its own linkages-between-words representation does not extend well to its constituent output, which was built as an afterthought to the parser, to attract users of government and binding grammars (Chomsky, 1981) to use the Link Grammar Parser. For one reason or another, a partial failure of the linkages representation, leads to a much more serious failure of the constituent output. One reason for this may be that the linkages representation is rather flat, in that the presence or absence of a linkage does not affect the presence or absence of other linkages. However, because constituent structure is hierarchical, a failure to recognize a constituent, which is a parent of other constituents, can trigger the failure of dependent (parent-child relationship) constituents to be recognized.

Another trouble with the Link Grammar Parser is that it was meant for clean and formal English, with all proper punctuation, and capitalization. Unfortunately, the email domain is not always clean and formal. Since Link Grammar Parser uses capitalization

and punctuation as recognition cues, the absences of these cues adversely affects the performance of the parser.

A third issue with Link Grammar Parser is that it does not fail in the right way. With government and binding grammar parsers that use part-of-speech information, the most basic syntactic constituent, the NP (noun phrase), is usually the first to be successfully recognized. In fact, the noun phrase recognition (also called noun phrase chunking) performance of such grammars is above 90%. Most failures come from the inability to combine NPs with other structures to build most complicated constituents. Unfortunately, in link grammars, its noun phrase chunking performance is not more reliable than the parser as a whole. The problem this presents for WALI is that NPs are arguably the most important constituents to recognize. NPs can be recognized as PERSON, PEOPLE, PLACE, TIME, EVENT, and THING by the semantic processing module; however, the recognition of the NP is contingent on the condition that the NP is recognized in the *first* place, by the syntactic parser. The bottom line is that failures to recognize NPs in syntactic parsing cause many annotations to be missed. Because of the importance of NP detection, it may be desirable in future implementations to use another syntactic parser that performs better on NP detection, or to run a separate NP back-off detection program. The best NP chunkers, such as that of Ramshaw and Marcus (1995), are able to achieve a respectable accuracy of 90-94%.

Another module that presents limitations to the performance of WALI is the part-of-speech tagger. In implementing the tagger in Java to be based on Brill's original transformation-based learning tagger, we used the original lexicon from Brill's UNIX implementation to guess the initial tag for a word. However, we do not have all the

context rules that Brill had in his implementation, which numbered in a couple of hundred. Training on 40,000 words of the Wall Street Journal corpus (Linguistic Data Consortium), we were only able to use the top 50 context rules. Though the performance of our part-of-speech tagger has not been formally evaluated, we can estimate it to perform about 10-15% worse than the state-of-the-art taggers, none of which are publicly available on Windows C++ or Java. We estimate the accuracy of our tagger to be a very poor 80% on any single word. The implication is that for a 10-word sentence, the probability that there are no incorrect tags is only 11%. Incorrect tags hurt the performance of the semantic processing module, because part-of-speech information is a component in some rules. The adverse effect of this tagger on the overall outcome of WALI is difficult to quantify. This limitation is quite easy to fix, however, in future implementations.

A third issue is the knowledge limitation of the understanding agents. A lexicon of 2,000 world-semantic concepts, each semantically categorized, may *seem* like a lot, but when we consider that the everyday world has hundreds of thousands of distinct concepts than we realize the system does not know enough. Additionally, the Open Mind Commonsense corpus does not assure an even distribution of the 2,000 concepts over a diverse range of concepts, so many topic areas, especially those not in the realm of commonsense knowledge, may not be covered. Thankfully OMCS is a public effort at gathering commonsense, and as its corpus grows, so will the knowledge of our system. There is also room for improvement on the way that knowledge is mined out of OMCS. Currently, the 2,000 concepts is only 0.5% of the 400,000 sentences in OMCS. The reason why we could only extract 2,000 concepts is because we could only use sentence

patterns whose blanks were properly semantically constrained. If we can improve the heuristics we use to identify concepts, we may be able to extract a much larger number of concepts. Another place to look for knowledge is the largest of the corpora – the Web. By parsing corpora similar to the text storytelling domain of ARIA, such as personal web logs like www.blogger.com, we may be able to not only extract many more concepts, but also have confidence that they are the right kinds of concepts, from a similar corpus-domain.

A fourth issue is that the class of learnable annotations is limited to the implemented semantic categories of PERSON, PEOPLE, PLACE, EVENT, THING, DATE/TIME, and EMOTION. There is also a need to implement a semantic category for ACTIVITY, which, unlike all of the currently implemented categories (except for EMOTION), is not NP-based. ACTIVITY is VP (verb-phrase) based. It is possible to create a dictionary of activities by mining them out of OMCS or some other resource. Members of the type ACTIVITY might include "kissing," "skiing," "baking," etc. This is immediately implementable using the current approach. What is *not* currently implementable are concepts that are a combination of two or more types. This is discussed in the section of the limitations of the approach.

Though limitations of the current implementation prevent the full potential of the approach from being realized, there are also more fundamental limitations, associated with the overall approach to semantic understanding taken in this thesis. We go on to discuss a few.

2.5.2 Limitations of the Approach

We discuss two major limitations to the semantic understanding approach taken in this thesis. 1) The types of annotations that can be learned using this approach are restricted to concepts that fall under a single category of the semantic categories (e.g. PERSON, PLACE, EVENT, THING, DATE/TIME, or EMOTION). 2) Building event structure information on top of syntactic constituents causes the resulting structure to be prone to a specific surface realization or verb alternation.

Compound Concepts

The current approach only allows for the learning of concepts, which fall under one and only one of the semantic categories, and cannot handle compound concepts – concepts that combine a verb with a NP or PP. NP-based concepts, such as PERSON, PLACE, EVENT, THING, and DATE/TIME are learnable in our system. AdjP-based concepts like EMOTION are also learnable. The system can also be extended to handle single verb ACTIVITY types, such as "reading," "skiing," etc. However, concepts which involve some combination of concepts are more difficult under our system's approach. For example, consider the text, "Here is a picture of Jane walking down the aisle." "Walking down the aisle" is clearly the subject of the photo and should be included as an annotation. The current system implementation might include "aisle" as an annotation. By adding a new semantic category for recognizing actions, we can also figure out that "walking" is an annotation. However, "walking down the aisle" should really be a single concept, and when it is fragmented into a verb and a NP, its meaning changes. Examples like "walking down the aisle" are more naturally a single concept because they are

somewhat idiomatic, or at least, the verb and object collocate with each other more often than usual.

One way to fix our approach to handle such phenomenon would be to create a knowledge base of all such idiomatic phrases. Although this would handle idiomatic concepts, it would not handle compound concepts that are not purely idiomatic, but do collocate with each other enough so that it would be desirable to link the concepts together. For example, "bobbing for apples" shows some signs of being a concept in its own right, and should not simply generate "bobbing" and "apples" as separate annotations.

To be able to understand that a linkage exists, we need verb selectional preferences knowledge, and verb-argument structure knowledge, to decide which objects or indirect objects of a verb *are* linked to the verb, and which should be considered independent annotations. Connecting the annotations "bobbing" and "apples" is beneficial to retrieval, because it would prevent false positive retrievals. For example, it might prevent a query for "head-bobbing" from retrieving "bobbing for apples." Just as it does not make sense to separate "San Francisco" into two annotations, "San" and "Francisco," it also does not make sense to separate verb and NP concepts that are logically a single ACTIVITY concept, such as "riding a bike," and "cutting a cake."

Event Structure on Syntactic Structure

Though there are several reasons why building event structure over syntactic structure is desirable (for one, the robustness of the syntactic structure), there is a tradeoff involved. For more robustness in obtaining that event structure, we must put in more effort, trying to use that structure. Most deep semantic parsers like UNITRAN and Cyc-

NL are interlinguas of sorts, and are quite independent of syntax. The reason they are able to do this is that they make use of verb-argument structure, and knowledge about the different alternations (a change in the surface realization of a verb-argument structure) for each verb. WALI however, does not do this because it is built on top of syntactic structures, which, by definition, encodes some syntactic information. More specifically, reading information off of WALI's event structure requires knowledge about specific alternations. To make this more concrete, let us look at some alternations of concern.

1. ***Understood Reciprocal Object Alternation***.

   John met Mary at the movies ⇐⇒ John and Mary met at the movies.

   Comment: The personal commonsense learner might want to learn, for example, all the people John has met. However, to learn from both of the above alternations, it would need two rules – one where "John and Mary" are in the AGENT role for "meet," and another where "John" is the AGENT, and "Mary" is the PATIENT. Having to encode two rules for essentially the same sentence said in two different ways illustrates one pitfall of WALI.

2. ***Preposition Drop Alternation***

   John met Mary ⇐⇒ John met with Mary.

   Comment: Two more alternations that must be handled by rules.

3. ***'There' Insertion***

   The wedding was in San Francisco ⇐⇒ There was a wedding in San Francisco.

   Comment: Although this does not present any problems to the annotation extraction, it might present a problem to the personal commonsense learner.

In the example, “wedding” switched from being the subject to the object of the verb ‘to be’.

In addition to the above verb alternations, there is one grammatical construction that can be an alternation to a large majority of verbs, and causes major woes for WALI. This is the nominalization of a verb, sometimes associated with the cheater verb, “to do.” For example, the sentence, “I visited London” can have the alternation, “I did some visiting of London.” By inserting the verb “to do” the main verb was nominalized. This presents a problem for the personal commonsense learner if it tried to track all the places the user has visited. Instead of being able to search for events whose main verb is “visit,” “went,” etc., it must also now search for nominalized verbs, which are most likely recognized by WALI as a THING. It might be possible to improve WALI for this specific grammatical construction by having it recognize nominalizations as verbs. However, this would be a hack rather than an eloquent solution.

Chapter 3

Commonsense Reasoning for Robust Photo Retrieval

Why is it that when a person is shown a picture of a flower girl dressed in white, she assumes it is at a wedding, and she wonders who is getting married? It is interesting that when presented with a scene or situation, people almost reflexive draw a host of conclusions. Just as interesting is that people generally draw the same conclusions about everyday scenes and situations such as the examples given above. Thinking about why this is the case, we conclude that people tend to draw similar conclusions because people share a "common sense" about the everyday world. Commonsense tells us to bring a present to a birthday party, to get medicine and rest when we feel sick, and to answer the door when we hear the doorbell. In short, it is a part of our everyday lives. When we communicate and interact with people, we assume that they possess commonsense, and they make a similar assumption about us. Whenever we do tasks in the everyday world, commonsense is indispensable. This begs the question: If we want a computer to help us with tasks in the everyday world, shouldn't we give the computer some commonsense too?

Our task of photo annotation and retrieval is indeed a task in the everyday world in which commonsense is indispensable. Imagine searching through a shoebox of pictures, looking for pictures of a wedding. We come across a picture of a woman in a white dress and a man in a black tuxedo, standing in front of a minister. We are sure that we have found a picture of a wedding. But how do we know? After all, there is no banner in the picture with a caption reading: "This is a Wedding!" We know because there is commonsense in our heads, which help us recognize that the woman in the white dress is a bride, the man wearing the black tuxedo is the groom, and that the bride and groom and priest are participants in a wedding ceremony. In this case, a human annotator/retriever used commonsense to find a photo. How would this apply to a computer program presented with the same task? Let's say that the user typed the word "wedding" into ARIA. Like in the physical shoebox, no pictures in ARIA's virtual shoebox had the annotation "wedding", but instead, let's assume that that the same photo mentioned above had the annotations "white dress", "tuxedo", "priest" associated with it. Without commonsense, a computer program would have a difficult time identifying this as a wedding photo. However, if our program had commonsense, it could reason in the same way as a person would, namely, that brides may wear white dresses; grooms may wear tuxedos; and brides, grooms, and priests are participants in a Christian wedding ceremony. (As you may have suspected, much of commonsense is culturally specific).

Having established that commonsense can be useful to our application, how do we go about giving our application commonsense? How much and what types of commonsense knowledge are needed? How is our knowledge represented, and how can we reason over

it in an efficient and robust way? This chapter presents an associative commonsense reasoning mechanism that can help to make retrieval in ARIA more robust.

This chapter is organized as follows. First, we begin by discussing how the world semantics of photos can be exploited to make retrieval by annotations more robust. Second, we discuss the nuances of the Open Mind Commonsense corpus that will be used in the implementation of ARIA's commonsense subsystem. Third, we describe the qualities of associative reasoning, a desirable mechanism for our problem domain. Fourth, we present the implementation of CRIS (Commonsense Robust Inference System), ARIA's commonsense service provider. Fifth, we discuss special cases of reasoning in ARIA, above and beyond what CRIS provides. Sixth, we conclude by discussing the limitations of the implementation of CRIS, and of the approach in general.

3.1 Exploiting World Semantics of Photos

There are many regularities to life in the everyday world, which take the form of familiar objects, well-known places, common social functions, and so on. These regularities – facts about the everyday world that seem almost "obvious," constitute our common understanding of the everyday world. We sometimes refer to such knowledge as "commonsense," but another way to describe this knowledge, as it pertains to language understanding, is world semantics. Semantics is the understanding of meaning, and so world semantics implies the understanding of a word or concept, with respect to how it fits into the world and interacts with other concepts from the world.

Organizing photos is a domain that can benefit from world semantic knowledge, because photos are snapshots of the world, and the structure of the world, and the

behavior of the people in the world, can be somewhat predictable, given some seed knowledge about the photo. We are not trying to attack the idea of free will, but we are merely pointing out that photos often depict people and things in common social situations, in well-known or common places.

If John were to tell Mary that he had a picture of someone surfing, she might guess that it was taken at the beach, that surfing is a sport, that there is an ocean wave, that the surfer is a person, wearing a wet suit, on a surfboard. It is not impossible that the surfer was somewhere else, -that there was not a beach, an ocean wave, wet suit, or surfboard; but it is likely that Mary guessed correctly, for the most part. Giving the knowledge that someone like Mary has to ARIA can be a great step toward robust retrieval. Suppose that in ARIA's shoebox, there was a picture annotated with surfer. Given Mary's world semantic knowledge, ARIA will be able to retrieve the photo when the user talks about the beach or about sports. Being able to make the semantic connection between those related concepts makes for an intelligent agent.

With that scenario as a goal, the approach we take to commonsense in ARIA is to use shallow commonsense reasoning techniques to associate semantically connected concepts with each other. The most direct way to do this is by expanding existing annotations using a world semantic resource. Such a resource needs to not only provide for associative reasoning, but must be efficient because of the scale of the knowledge involved. Before we discuss how such a resource can be constructed, and used to reason with, we should first examine the capabilities of the corpus of commonsense knowledge we will be using. The qualities and quantities of knowledge we are able to extract depend heavily on the representation and the characteristics of the corpus. The next

section focuses discussion on the Open Mind Commonsense corpus, and some comparison to a rival commonsense corpus, Cyc, is presented.

3.2 OMCS: A Corpus of Commonsense

The source of the world semantic knowledge used by our mechanism is the Open Mind Commonsense Knowledge Base (OMCS) (Singh, 2002) - an endeavor at the MIT Media Laboratory that aims to allow a web-community of teachers to collaboratively build a database of "common sense" knowledge. Web teachers do not need formal training in logic or any other training, because users impart commonsense knowledge to Open Mind by writing English sentences and short stories by filling in the blanks.

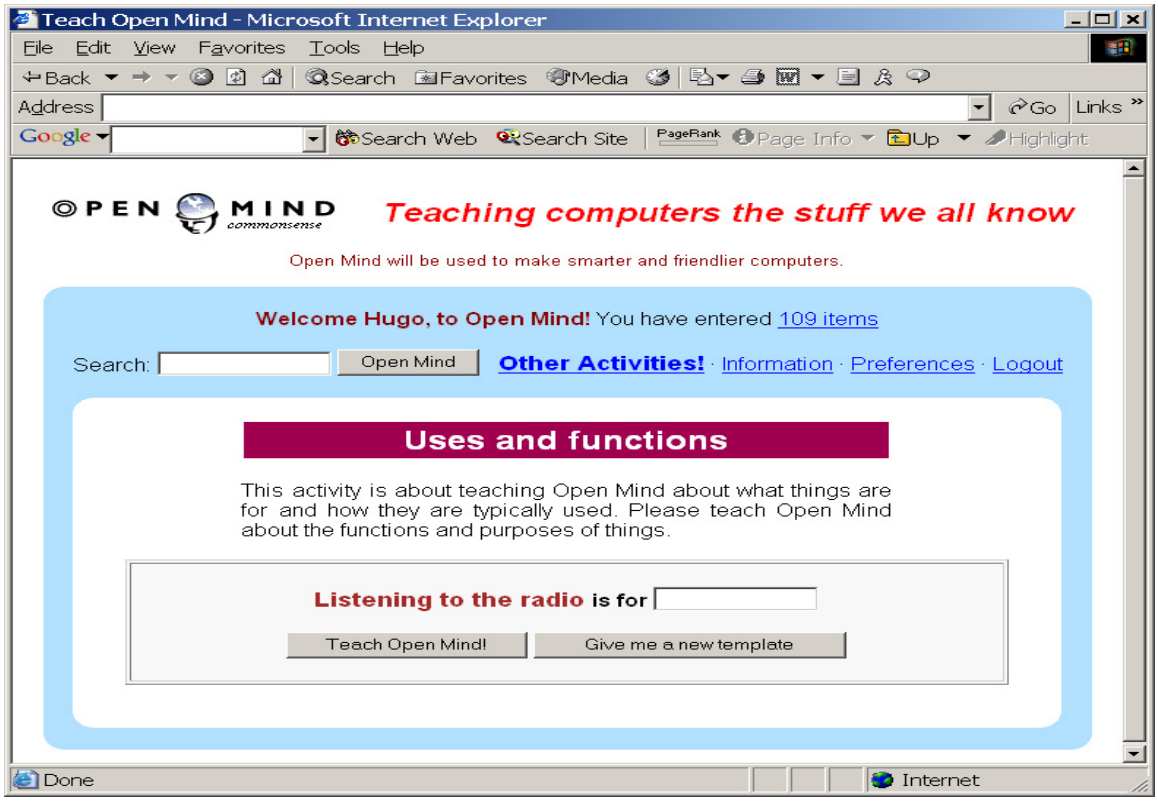

Figure 8: Screenshot of an activity in Open Mind. Users simply fill-in-the-blank to teach Open Mind some commonsense.

Figure 8 shows a screenshot of one of twenty activities in OMCS, each of which imparts a different commonsense relation. Note that the sentence in Figure 8 has two blanks. One blank, i.e. "listening to the radio," was filled in by OMCS. The user is requested to fill in the other blank.

It is hard to define what actually constitutes commonsense, but in general, one can think of it as knowledge about the everyday world that most people within a certain society consider to be "obvious." It is important to point out that any semantic resource that attempts to encapsulate common knowledge about the everyday world is going to be somewhat culturally specific. The main example used in this thesis of weddings having brides, grooms, wedding gowns, bridesmaids, flower girls, and churches illustrates an important point: knowledge that is *obvious* and *common* to one group of people (in this case, middle-class USA) may not be so obvious or common to other groups. Although many thousands of people from around the world collaboratively contribute to Open Mind Commonsense, the majority of the knowledge in the corpus reflects the cultural bias of middle-class USA. In the future, it may make sense to gather commonsense for multiple cultures, and tag knowledge by their cultural specification.

In presenting OMCS, we give an overview of the corpus, compare it with its more famous counterpart, Cyc, and talk about how noise and ambiguity must be managed in order for OMCS to be useful to our purposes.

3.2.1 A Corpus Overview

OMCS contains over 400,000 semi-structured English sentences about commonsense, the majority of which are organized into an ontology of commonsense relations such as the following:

- *A is a B*
- *You are likely to find A in/at B*
- *A is used for B*

By semi-structured English, we mean that many of the sentences loosely follow one of 20 or so sentence patterns in the ontology. However, the words and phrases represented by A and B (see above) are not restricted. Some examples of sentences in the knowledge base are:

- *Something you find in (a restaurant) is (a waiter)*
- *The last thing you do when (getting ready for bed) is (turning off the lights)*
- *While (acting in a play) you might (forget your lines)*

The parentheses above denote the part of the sentence pattern that is unrestricted. While English sentence patterns has the advantage of making knowledge easy to gather from ordinary people, there are also problems associated with this.

The major limitations of OMCS are four-fold. First, there is ambiguity resulting from the lack of disambiguated word senses, and from the inherent nature of natural languages. Second, many of the sentences are unusable because they may be too complex to fully parse with current parser technology. Third, because there is currently no truth maintenance mechanism or filtering strategy for the knowledge gathered (and such a mechanism is completely nontrivial to build), some of the knowledge may be anomalous,

i.e. not commonsense, or may plainly contradict other knowledge in the corpus. Fourth, in the acquisition process, there is no mechanism to ensure a broad coverage over many different topics and concepts, so some concepts may be more developed than others.

At the time of the writing of this thesis, a next-generation Open Mind Commonsense is being built that will hopefully address many of the ambiguity and noise limitations of the original OMCS.

3.2.2 OMCS versus Cyc

The Open Mind Commonsense Knowledge Base is often compared with its more famous counterpart, the Cyc Knowledge Base (Lenat, 1998). Cyc contains over 1,000,000 hand-entered rules that constitute "common sense". Unlike OMCS, Cyc represents knowledge using formal logic, and ambiguity is minimized. In fact, it does not share any of the limitations mentioned for OMCS. Of course, the tradeoff is that whereas a community of non-experts contributes to OMCS, Cyc needs to be somewhat carefully engineered. Unfortunately, the Cyc corpus is not publicly available at this time, whereas OMCS *is* freely available and downloadable via its website (www.openmind.org/commonsense).

3.2.3 Managing Ambiguity and Noise

Even though OMCS is a more noisy and ambiguous corpus, we find that it is still suitable to our task. By normalizing the concepts, we can filter out some possibly unusable knowledge. The impact of ambiguity and noise can be minimized using heuristics that account for these factors. Even with these precautionary efforts, some

anomalous or bad knowledge will still exist, and can lead to seemingly semantically irrelevant concept expansions. In this case, we rely on the fail-soft nature of the ARIA application that uses this semantic resource to handle noise gracefully.

3.3 Associative Commonsense Reasoning

The commonsense reasoning mechanism we propose is an associative one. In order to understand the similarities and differences of our approach to more traditional reasoning mechanisms, we first examine the very standard method of first-order inference.

First-Order Inference

Traditionally, reasoning is thought of drawing inferences from known facts. For example, given facts A and B, conclusion C might be drawn:

A: *Brides are part of weddings*

B: *Weddings usually take place in churches*

C: *Brides are often found in churches*.

In the above example, the rule that allowed for inference C to be made was *transitivity*. That is to say,

(A Relation$_1$ B) and (B Relation$_2$ C) ➔ (A Relation$_3$ C)

The vast majority of first order inferences invoke some kind of transitivity relationship. The hard part about making this inference is, given what Relation$_1$ and Relation$_2$ are, how can we choose the correct Relation$_3$ which accurately connects A and C? One solution is to collect all combinations of Relation$_1$ and Relation$_2$ and Relation$_3$ that produce valid inferences. These are called inference patterns. However, there is a

further complication because the validity of an inference pattern also depends on the semantic categories of A, B, and C. (This is assuming that it is impossible to eliminate ambiguity in defining all Relations.)

While we gave only one step of an inference, most inferences occur in chains, over many steps of reasoning. Usually, there is a goal for reasoning, which is often to answer a question, solve a problem, or draw a conclusion. When there is a goal, there must also be a specific path or paths that will lead us from our initial assumptions to some final state. Unfortunately, finding this path, known as directed inference, is incredibly difficult at best. Brute force search across all assertions in the knowledge base can lead to the oft-dreaded combinatorial explosion of search space.

The discovery of inference patterns and combinatorial explosion of directed inference are two of the tradition pitfalls of reasoning. However, as we discuss in the following section, these pitfalls can be avoided in the associative reasoning approach to commonsense in the photo annotation domain.

Associative Reasoning

In ARIA's photo annotation and retrieval domain, commonsense reasoning is needed to relate existing annotations to concepts in the text pane through some level of semantic connectedness. For example, the concept "wedding" in the text should trigger a photo with the annotation "bride." When thinking about the mechanism that might accomplish this association, there are two solutions. In the first solution, depicted in Figure 9, two concepts are fed into an "oracle" which would return either a binary answer of relatedness or a percent score proportional to their relatedness.

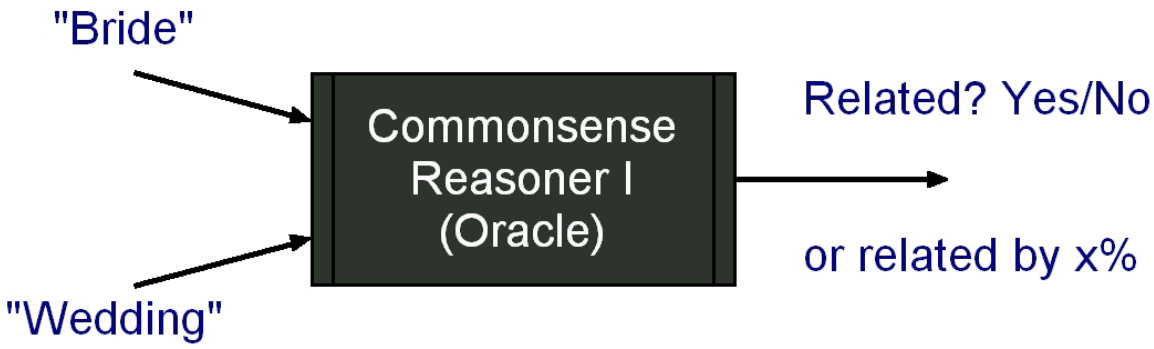

Figure 9: Black box reasoner candidate #1: The Oracle

It is easy to imagine how this model could fit into the framework of first-order inference. "Bride any_relation wedding" would be the inference goal. The mechanism would perform exhaustive search, chaining together all its knowledge to try to arrive at this goal. However, we want to avoid inference patterns and an exponential search space if possible.

It is possible to think of the problem in terms of a second mechanism. A purely associative reasoning mechanism that would not be subject to the pitfalls of traditional first-order inference. Consider the black box reasoner depicted in Figure 10.

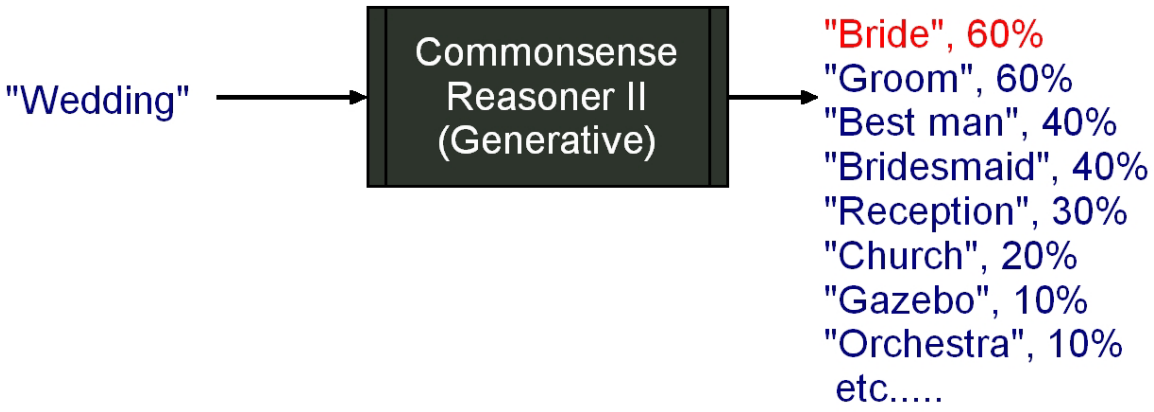

Figure 10: Black box reasoner candidate #2: Generative

Rather than taking in an inference goal of "wedding any_relation bride," in this model, we instead treat commonsense relations as ways for concepts to semantically connect, and we follow these connections using transitivity. This method is generative because it returns a list of all concepts closely related to "wedding," without any mention of how they are related. In fact, the relations are rather irrelevant in this type of associative mechanism. Since there are no relations, no inference pattern is needed because all inferences are valid. (Admittedly, there are different strengths associated with each relation. We discuss this in the implementation section). There is no combinatorial explosion because we can use the strength of a path in the search tree (the product of the strengths of each relation in the path) to prune the search. Put in a different way, semantic connectedness between concepts is measured by a confidence score, which is lowered every time a new inference is chained. Applying a threshold, or tolerance score, to the search is one way to avoid combinatorial explosion. Additionally, it may be desirable to restrict the number of nodes expanded at each level of the tree to the nodes with the highest confidence scores. This is often referred to as beam search, or *n-best* search.

This generative model of commonsense is the associative reasoning approach taken by this thesis. The inputs to the black box are existing annotations (either manually annotated, or learned by the automated annotation mechanism). A list of commonsense annotation expansions is returned, along with a confidence score for each entry. These expansions can be incorporated as new annotations, provided that their values are discounted, compared to non-commonsensical annotations. In the retrieval process,

concepts from the text will be able to match a much larger variety of concepts than just those explicit annotations.

Based on the requirements we have laid out for the associative reasoning mechanism (that it has an uncertainty model to measure the strength of associations, that search should be pruned using beam and threshold scores), the most appropriate data structure to represent this is a spreading activation network. In this network, nodes represent commonsense concepts, and edges between nodes represent semantic connectedness (commonsense relations), and are weighted with a confidence score. To expand an annotation, start from that annotation's concept node and visit all connected nodes one hop away, two hops, and so on. Nodes will only continue to activate if their activation energy meets a minimum score (threshold). It is also possible to limit the number of activated nodes at each level to the *n-best* (beam search).

In the following section, we introduce CRIS, a spreading activation network of commonsense for annotation expansion. Discussion focuses on how the spreading activation network is automatically constructed by mapping from knowledge in OMCS, and how noise and ambiguity are heuristically adjusted for in the network.

3.4 CRIS: Commonsense Robust Inference System

The Commonsense Robust Inference System (CRIS) is an agency for associative commonsense reasoning. It takes as input a semantically typed photo annotation from ARIA, such as a "wedding" (EVENT), or "California" (PLACE), and expands the input annotation by means of a spreading activation network, using knowledge that is

automatically extracted from the Open Mind Commonsense database. CRIS returns a collection of additional annotations garnered through commonsense expansion, along with a confidence score for each annotation. CRIS encompasses both the mechanism for automatically constructing the commonsense spreading activation network, and the mechanism, which uses that network to perform reasoning. CRIS can be decomposed into the following steps:

1. Extracting predicate argument structure from semi-structured English
2. Modeling commonsense concepts and relations as a directed concept node graph
3. Commonsense inference as spreading activation search over the graph

Auxiliary tools and data structures were built to support the above mechanisms, including the following:

1. Part-of-speech tagger
2. Word-form Lemmatiser
3. Phrase to Concept Mapper
4. Miniature Noun Phrase and Activity Phrase Grammar
5. Predicate argument structure mapping rules
6. Uncertainty model
7. Node Reinforcement Heuristic
8. Inverse Popularity Heuristic

Robustness

CRIS was designed to address the issues of robustness and efficiency. The Phrase to Concept Mapper adds robustness to CRIS's representation of commonsense by allowing

annotation phrases to "fuzzy match" commonsense concepts. Thinking about the requirements of the larger system ARIA, CRIS employs an uncertainty model to return the most relevant annotation expansions possible, but in cases where the commonsense knowledge is sparse, CRIS can still return many, albeit less relevant, annotation expansions. From the user interface perspective, this provides an enhanced degree of robustness. In cases where the search keywords and photo annotations can't be easily linked through commonsense, CRIS will attempt to weakly link keywords to photo annotations by exploring more levels of inference.

Another aspect of robustness is having to deal with noise in the commonsense knowledge. OMCS can be noisy. Noise arises because of rogue knowledge, knowledge that is not commonsense, and from ambiguous word senses. With the fairly shallow techniques used to parse commonsense facts into predicate argument structure, it is difficult to perform word-sense disambiguation. Therefore a shortcoming of the Concept Node Graph is that the concept nodes are not tagged with word senses. To cope with this problem, a Node Reinforcement Heuristic is applied to the graph to group concepts that are all mutually related. This heuristic gives an approximation for word senses. In addition, an Inverse Popularity Heuristic is used to focus the spreading activation search by activating more specific concept nodes (less popular) first. These two heuristics are ways to deal with noise given the limitations of the source of commonsense knowledge, and the shallow techniques used to produce the graph.

Efficiency

Efficiency is a big issue in ARIA because inference needs to be performed in real-time, as the user types an email. Traditionally, inferences over very large knowledge

bases are not known to be very efficient because of the exponential blowup in search space. The spreading activation network data structure is rather efficient, considering the amount of knowledge involved. Implemented as a large hash table whose keys are nodes and values are arrays of pointers pointing to connected nodes, along with weights, the runtime of a single annotation expansion is on the order of 0.25 seconds, running on a 1 GHz Wintel box.

3.4.1 Building a World Semantic Resource

In this section, we describe how a usable subset of the knowledge in OMCS is extracted and structured specifically for the photo retrieval task. First, we apply sentence pattern rules to the raw OMCS corpus and extract crude predicate argument structures, where predicates represent commonsense relations and arguments represent commonsense concepts. Second, concepts are normalized using natural language techniques, and unusable sentences are discarded. Third, the predicate argument structures are read into a Concept Node Graph, where nodes represent concepts, and edges represent predicate relationships. Edges are weighted to indicate the strength of the semantic connectedness between two concept nodes.

Extracting Predicate Argument Structures

The first step in extracting predicate argument structures is to apply a fixed number of mapping rules to the sentences in OMCS. Each mapping rule captures a different commonsense relation. Commonsense relations, insofar as what interests us for constructing our world semantic resource for photos, fall under the following general categories of knowledge:

1. Classification: A dog is a pet
2. Spatial: San Francisco is part of California
3. Scene: Things often found together are: restaurant, food, waiters, tables, seats
4. Purpose: A vacation is for relaxation; Pets are for companionship
5. Causality: After the wedding ceremony comes the wedding reception.
6. Emotion: A pet makes you feel happy; Rollercoasters make you feel excited and scared.

In our extraction system, mapping rules can be found under all of these categories. To explain mapping rules, we give an example of knowledge from the aforementioned Scene category:

```
somewhere THING1 can be is PLACE1
somewherecanbe
THING1, PLACE1
0.5, 0.1
```

Mapping rules can be thought of as the grammar in a shallow sentence pattern-matching parser. The first line in each mapping rule is a sentence pattern. THING1 and PLACE1 are variables that approximately bind to a word or phrase, which is later mapped to a set of canonical commonsense concepts. Line 2 specifies the name of this predicate relation. Line 3 specifies the arguments to the predicate, and corresponds to the variable names in line 1. The pair of numbers on the last line represents the confidence weights given to forward relation (left to right), and backward relation (right to left), respectively, for this predicate relation. This also corresponds to the weights associated with the directed edges between the nodes, THING1 and PLACE1 in the graph representation.

It is important to distinguish the value of the forward relation on a particular rule, as compared to a backward relation. For example, let us consider the commonsense fact, *"somewhere a bride can be is at a wedding."* Given the annotation *"bride,"* it may be very useful to return *"wedding."* However, given the annotation *"wedding,"* it seems to be less useful to return *"bride," "groom," "wedding cake," "priest,"* and all the other things found in a wedding. For our problem domain, we will generally penalize the direction in a relation that returns hyponymic concepts as opposed to hypernymic ones. The weights for the forward and backward directions were manually assigned based on a cursory examination of instances of that relation in the OMCS corpus.

Approximately 20 mapping rules are applied to all the sentences (400,000+) in the OMCS corpus. From this, a crude set of predicate argument relations are extracted. At this time, the text blob bound to each of the arguments needs to be normalized into concepts.

Concept Refinement Grammar

Because any arbitrary text blob can bind to a variable in a mapping rule, these blobs need to be normalized into concepts before they can be useful. There are three categories of concepts that can accommodate the vast majority of the parseable commonsense knowledge in OMCS: Noun Phrases (things, places, people), Attributes (adjectives), and Activity Phrases (e.g.: *"walk the dog," "buy groceries."*), which are verb actions that take either no argument, a direct object, or indirect object.

To normalize a text blob into a Noun Phrase, Attribute or Activity Phrase, we tag the text blob with part of speech information, and use these tags filter the blob through a miniature grammar. If the blob does not fit the grammar, it is massaged until it does or it

is rejected altogether. Sentences, which contain text blobs that cannot be normalized, are discarded at this point. The final step involves normalizing the verb tenses and the number of the nouns. Only after this is done can our predicate argument structure be added to our repository.

The aforementioned noun phrase, and activity phrase grammar is shown below in a simplified view. Attributes are simply singular adjectives.

```
NOUN PHRASE:
 (PREP) (DET|POSS-PRON) NOUN
 (PREP) (DET|POSS-PRON) NOUN NOUN
 (PREP) NOUN POSS-MARKER (ADJ) NOUN
 (PREP) (DET|POSS-PRON) NOUN NOUN NOUN
 (PREP) (DET|POSS-PRON) (ADJ) NOUN PREP NOUN

ACTIVITY PHRASE:
 (PREP) (ADV) VERB (ADV)
 (PREP) (ADV) VERB (ADV) (DET|POSS-PRON) (ADJ) NOUN
 (PREP) (ADV) VERB (ADV) (DET|POSS-PRON) (ADJ) NOUN NOUN
 (PREP) (ADV) VERB (ADV) PREP (DET|POSS-PRON) (ADJ) NOUN
```

The grammar is used as a filter. If the input to a grammar rule matches any optional tokens, which are in parentheses, then this is still considered a match, but the output will filter out any optional fields. For example, the phrase, *"in your playground"* will match the first rule and the phrase will be stripped to just *"playground."*

Concept Node Graph

To model concept expansion as a spreading activation task, we convert the predicate argument structures gathered previously into a Concept Node Graph by mapping arguments to concept nodes, and predicate relations to edges connecting nodes. Forward

and backward edge weights come from the mapping rule associated with each predicate relation. A segment of the graph is shown in Figure 11.

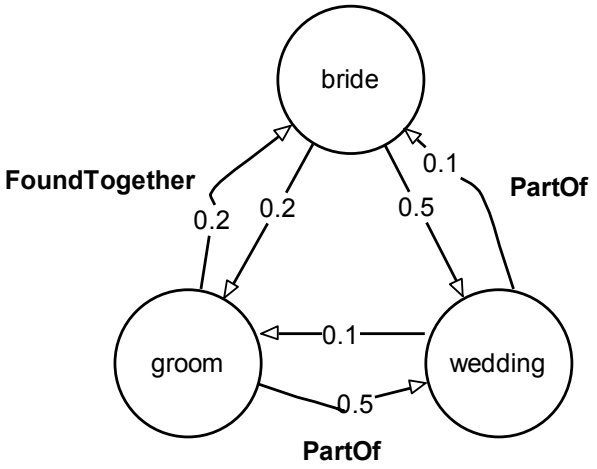

Figure 11: A portion of the Concept Node Graph. Nodes are concepts, and edges correspond to predicate relations.

The following statistics were compiled on the automatically constructed resource:

- 400,000+ sentences in OMCS corpus
- 50,000 predicate argument structures extracted
- 20 predicates in mapping rules
- 30,000 concept nodes
- 160,000 edges
- average branching factor of 5

3.4.2 A Spreading Activation Network

In this section, we explain how concept expansion is modeled as spreading activation. We propose two heuristics for re-weighting the graph to improve relevance. Examples of the spreading activation are then given.

In spreading activation, the origin node is the concept we wish to expand (i.e. the annotation) and it is the first node to be activated. Next, nodes one hop away from the origin node are activated, then two levels away, and so on. A node will only be activated if its activation score (AS) meets the activation threshold, which is a tolerance level between 0 (irrelevant) and 1.0 (most relevant). The origin node has a score of 1.0. Given two nodes A and B, where A has 1 edge pointing to B, the activation score of B is given in equation (1).

$$AS(B) = AS(A) * weight(edge(A,B)) \qquad \textbf{(1)}$$

When no more nodes are activated, we find all the concepts that expand the input concept up to our set threshold.

Heuristics for Coping with Noise

One problem that can arise with spreading activation is that nodes that are activated two or more hops away from the origin node may quickly lose relevance, causing the search to lose focus. One reason for this is noise. Because concept nodes do not make distinctions between different word senses (an aforementioned problem with OMCS), it is possible that a node represents many different word senses. Therefore, activating more than one hop away risks exposure to noise. Although associating weights with the edges

provides some measure of relevance, these weights form a homogenous class for all edges of a common predicate (recall that the weights came from mapping rules).

We identify two opportunities to re-weight the graph to improve relevance: reinforcement and popularity. Both of these heuristics are known techniques associated with spreading activation networks (Salton and Buckley, 1988). We motivate their use here with observations about our particular corpus, OMCS.

Reinforcement Heuristic

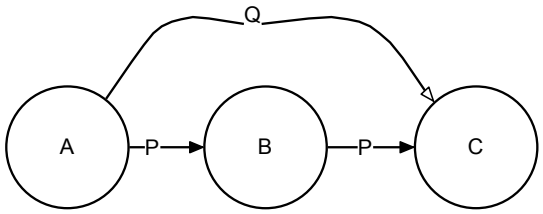

Figure 12: An example of reinforcement

As illustrated in Figure 12, we make the observation that if node C is connected to node A through both paths P and Q, then C would be more relevant to A than had either path P or Q been removed. We call this *reinforcement* and define it as two or more corroborating pieces of evidence, represented by paths, that two nodes are semantically related. The stronger the reinforcement, the higher the potential relevance.

Looking at this in another way, if three or more nodes are mutually connected, they form a cluster. Examples of clusters in our corpus are higher-level concepts such as weddings, sporting events, parties, etc., each with many inter-related concepts associated with them. Within each such cluster, any two nodes have enhanced relevance because the other nodes provide additional paths for reinforcement. Applying this, we re-weight the graph by detecting clusters and by increasing the weight on edges within the cluster.

Popularity Heuristic

The second observation we make is that if an origin node A has a path through node B, and node B has 100 children, then each of node B's children is less likely to be relevant to node A than if node B had had 10 children.

We refer to nodes with a large branching factor as being popular. It happens that popular nodes in our graph tend to either correspond to very common concepts in commonsense, or tend to have many different word senses, or word contexts. This causes its children to have, in general, a lower expectation of relevance.

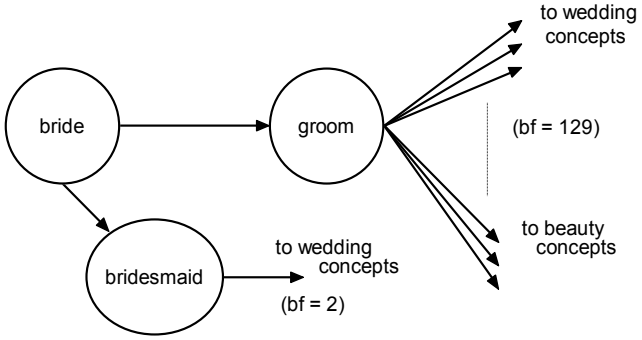

Figure 13: Illustrating the negative effects of popularity

As illustrated in Figure 13, the concept *bride* may lead to *bridesmaid* and *groom*. Whereas *bridesmaid* is a more specific concept, not appearing in many contexts, *groom* is a less specific concept. In fact, different senses and contexts of the word can mean "the groom at a wedding," or "grooming a horse" or "he is well-groomed." This causes *groom* to have a much larger branching factor.

It seems that even though the nature of our knowledge is commonsense, there is more value associated with more specific concepts than general ones. To apply this principle, we visit each node and discount the weights on each of its edges based on the metric in equation (2). (α and β are constants):

$$newWeight = oldWeight * discount$$
$$discount = \frac{1}{\log(\alpha * branchingFactor + \beta)} \quad \textbf{(2)}$$

Examples and Limitations

Below are actual runs of the concept expansion program using an activation threshold of 0.1. They were selected to illustrate what can be commonly expected from the expansions, including limitations posed by the knowledge.

```
>>> expand("bride")
('love', '0.632'), ('wedding', '0.5011')
('groom', '0.19'), ('marry', '0.1732')
('church', '0.1602'), ('marriage', '0.1602')
('flower girl', '0.131') ('happy', '0.131')
('flower', '0.131') ('lake', '0.131')
('cake decoration', '0.131') ('grass', '0.131')
('priest', '0.131') ('tender moment', '0.131')
('veil', '0.131') ('wife', '0.131')
('wedding dress', '0.131') ('sky', '0.131')
('hair', '0.1286') ('wedding bouquet', '0.1286')
('snow covered mountain', '0.1286')

>>> expand('london')
('england', '0.9618') ('ontario', '0.6108')
('europe', '0.4799') ('california', '0.3622')
('united kingdom', '0.2644') ('forest', '0.2644')
('earth', '0.1244')

>>> expand("symphony")
('concert', '0.5') ('music', '0.4')
('theatre', '0.2469')
('conductor', '0.2244')
('concert hall', '0.2244')
('xylophone', '0.1') ('harp', '0.1')
('viola', '0.1') ('cello', '0.1')
('wind instrument', '0.1') ('bassoon', '0.1')
('violin', '0.1')

>>> expand("listen to music")
('relax', '0.4816') ('be entertained', '0.4816')
('have fun', '0.4') ('relaxation', '0.4')
('happy', '0.4') ('hang', '0.4')
('hear music', '0.4') ('dorm room', '0.4')
('understand', '0.4') ('mother', '0.2')
('happy', '0.136')
('get away', '0.136') ('listen', '0.136')
```

```
('change psyche', '0.136') ('show', '0.1354')
('dance club', '0.1295') ('frisbee', '0.1295')
('scenery', '0.124') ('garden', '0.124')
('spa', '0.124') ('bean bag chair', '0.124')
```

The expansion of *"bride"* shows the diversity of relations found in the semantic resource. *"Love"* is some appropriate emotion that is implicitly linked to brides, weddings, and marriage. Expansions like *"priest", "flower girl,"* and *"groom"* are connected through social relations. *"Wife"* seems to be temporally connected. To *"marry"* indicates the function of a wedding.

However, there are also expansions whose connections are not as obvious, such as *"hair,"* and *"lake."* There are also other expansions that may be anomalies in the OMCS corpus, such as *"tender moment"* and *"snow covered mountain."* These examples point to the need for some type of statistical filtering of the knowledge in the corpus, which is not currently done.

In the last expansion example, the concept of "listen to music" is arguably more abstract than the wedding concept, and so the expansions may seem somewhat arbitrary. This illustrates one of the limitations of any commonsense acquisition effort: deciding upon which topics or concepts to cover, how well they are covered, and to what granularity they are covered.

3.5 Special Case of Reasoning

The associative reasoning mechanism described thus far works by expanding a photo's annotations with commonsense. This is sufficient for many types of reasoning including social reasoning, emotional reasoning, reasoning about scenes, and reasoning about the purpose or function of a concept. However, there is at least one (possibly

more) special case of reasoning that not only makes use of the annotation to be expanded as input, but also takes in as input, contextual information about a photo, namely, the photo's timestamp. Combining this contextual knowledge with causal and temporal commonsense, it becomes possible to *cross-annotate* photos. That is to say, reasoning about one photo's annotation results in another photo being annotated. We consider this special case of causal reasoning below.

3.5.1 Causal

A subset of the knowledge in OMCS describes causal and temporal relationships between concepts. There are approximately 9,000 sentences in this subset, with sentences such as "Something that might happen shortly after a wedding ceremony is a wedding reception." Just as general knowledge is compiled into a spreading activation network, causal and temporal knowledge is also compiled into a miniature spreading activation network. In addition to a confidence score label on each edge in the network, there is also a time interval that describes the amount of time that should be elapsed in the casual relation from node A to node B. In the above example, defining "shortly after" as being from 0 to 5 hours apart, the above sentence will generate the following segment of the network, shown in Figure 14.

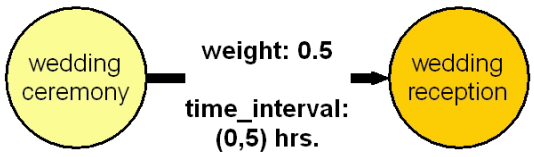

Figure 14: Segment of the Causal SAN showing a new edge feature

In chaining across nodes two or more hops apart, not only are the edge weights along multiplied, but the "time_interval" fields are also updated by adding the left endpoints of the intervals to get the left endpoint of the new interval, and similarly for the right endpoint. Using this causal spreading activation network, we can imagine the following cross-annotation scenario.

1. John is writing an email to a friend, describing Ken and Mary's wedding. He describes one wedding photo as: "The ceremony was fantastically well done" and proceeds to drag in the photo into the email.
2. In doing so, that picture gets annotated with "ceremony" and general commonsense expansions are performed, resulting in many commonsense annotations pertaining to weddings being added to that same photo.
3. Then the causal SAN runs and all photos taken in the timespan of 0 to 5 hours *after* the timestamp of the first photo receive the commonsense cross-annotation, "reception."
4. John continues writing his email and as he begins talking about the wedding reception, all the photos from the reception pop up. He is pleasantly surprised.

Cross annotation is a powerful way to demonstrate temporal commonsense. We have shown that it can use the same mechanism and data structure as general commonsense associative reasoning, with very little modification. There are also other types of reasoning which may benefit from contextual information about a photo and cross-annotation.

One leading example is geospatial commonsense reasoning. Many new high-end digital cameras are now equipped with global positioning systems (GPS) that can embed the GPS coordinates of a photo's location into the digital picture. Using this GPS-stamp, it is possible to plot the photo locations on a map. Then using conventional map databases of cities and landmarks, we can automatically annotate photos with their landmark location.

However, we see another possible application, involving geospatial commonsense. Whereas GPS coordinates easily pull up landmarks and monuments, they cannot typically pull up everyday places. For example, consider the concept, "snorkeling." Commonsense tells us that snorkeling often happens near an ocean reef, which is also close to a beach. We can combine this geospatial commonsense with GPS information to do commonsense cross-annotation as follows: Given a photo annotated with "snorkeling," GPS information can return all photos taken near the original photo. All those photos can be annotated with guesses of their location, such as "reef" and "beach."

3.6 Limitations

The concept-based, associative approach to commonsense reasoning is quite suitable to the task of expanding explicit annotations. Without image processing on photos or other contextual information about photos, the only pieces of evidence available are the concept annotations. Though expanding annotations using commonsense amounts to guessing, it is the best that can be done with so little information. And the approach does seem to improve the robustness of photo retrieval. In preliminary evaluations of the approach, commonsense expansions did help users connect concepts from the text to

semantically connected annotations on photos. In cases where commonsense suggested a photo that the user did not find to be relevant, the user simply ignored the suggestion and continued typing. Through the fail-soft nature of the ARIA photo suggestions, poor suggestions did not significantly frustrate or distract the user, who could continue pursuing the task.

Currently, there is only a small percentage of human commonsense available to the system. Without any claims about the quality of the knowledge in CRIS, the 30,000 concept nodes and 80,000 edge pairs still represent a minute fraction of commonsense that people possess. Minsky estimates that humans have on the order of 20 to 30 million pieces of commonsense knowledge. This prediction presents two problems: 1) the difficulty in acquiring such a vast and complete amount of knowledge, and 2) the computational limitations posed by wading through such knowledge.

Cyc has spent a decade and several million dollars to build a database of 1,000,000 pieces of knowledge, albeit, of rather high quality and acceptable coverage of topics. OMCS, a public acquisition effort, has acquired 400,000 English sentences in a couple of years, though currently the knowledge is plagued by noise, ambiguity, and coverage problems. However, if some of these problems can be addressed, the public acquisition of commonsense knowledge seems very promising.

Our current approach and all other attempts to impart a large amount of commonsense knowledge to computers will be computationally very difficult to perform in real-time. The current performance of the CRIS is averaging 0.25 seconds to expand one annotation on a 1 GHz Wintel machine. Increasing the knowledge in the hash table by three orders of magnitude would not only face memory limitations, but the amount of

processing time taken would prevent real-time expansions. There are two ways in which this computational limitation might be overcome. Either the memory capacities and processing capabilities will need to grow by several orders of magnitude (the brute-force solution), *or*, there will need to be a more intelligent, perhaps cognitively motivated, way to organize and prioritize knowledge, which has yet to be discovered by the fields of AI and Cognitive Science.

Chapter 4

Learning Personal Commonsense

Chapter 3 illustrates how commonsense knowledge might be used to make photo retrieval more robust. It does this by bridging the "semantic gap" between the keyword annotation and the keywords in the text by applying everyday knowledge about how the world works. As such knowledge is common and obvious to people in the real world, it is generally useful to users. In the field of Human-Computer Interaction (HCI), commonsense might be viewed as a generic user model that serves as the foundation for the user models of individual users. What, then, are specifications to the user model called? Just as commonsense knowledge is common and obvious to everyone, personal commonsense knowledge is knowledge that is obvious to the user, and people in the user's inner circle, such as close friends and family. We argue that just as commonsense corresponds to a generic user model, personal commonsense corresponds to specifications to that user model. As ordinary commonsense helps to connect semantically related concepts between annotations and the text, personal commonsense can serve a similar purpose, as well as having other applications.

This chapter is organized in four parts. First we present methods and types of personal commonsense that is learned by observing the user's text. Second we discuss current and future applications of personal commonsense to improving the performance

of the system. Third, we describe the implementation of the PCSL (Personal Commonsense Learner). Fourth, we conclude with a discussion of the limitations of the approach presented in this chapter.

4.1 Methods and Types of Personal Commonsense

To understand the nature of the personal commonsense ARIA hopes to learn, we need to start by considering the sorts of knowledge the system aims to learn about the user, and the methods the system hopes to employ for the learning mechanism.

4.1.1 Types of Personal Commonsense

The types of personal commonsense that can be induced through observational learning of the text have the potential to encompass a great variety, limited only by the level of understanding achievable by WALI's event structure parser. The current implementation focuses on two areas of personal commonsense: 1) places the user has been and events s/he has attended; and 2) understanding the user's relationships with familiar people.

The places the user has visited and the events s/he has attended constitute a "travel log" of sorts. This information might be useful in producing visualizations for information retrieval and other potential applications. Examples of this type of knowledge are expressed as personal commonsense as follows:

- The user was in San Francisco from 1/1/2002 to 1/5/2002.
- The user went to a wedding from 1/1/2002 to 1/5/2002

Another type of personal commonsense, with more immediate applications in terms of improving retrieval, has to do with the relationships between the user and people familiar to her. Knowing the names of the people in the user's every life, and their relationships to the user allows the system to make the semantic connection between a person's name and their relationship to the user in the retrieval process. For example, learning that the user has a sister named Mary will allow ARIA to connect the keyword "my sister" in the text to a photo annotated with "Mary." People who have "named" relationships to the user are not limited to family members. These named relationships can also be roles or attributes that the user commonly uses to describe a person. Roles and attributes might include things like "doctor," "Bill's wife," "best man," "classmate," "pharmacist," "friend," etc. Examples of this type of personal commonsense include:

- The user has a neighbor named John Smith.
- The user has a sister named Mary.
- The user has a friend named Jane Coldwell.

In addition to learning the named relationship between the user and a familiar person, the system also learns about familiar people and the context (physical and topical) that they are often talked about in. For example, a guy named "Mark" works in the user's office, and the user has written about Mark and the office in the same context a few times, using ARIA. ARIA can learn this useful piece of knowledge that Mark is often found in the office. Or suppose the example is about "Jane" who is often written about in connection with the topic, "rock climbing." ARIA can also associate "Jane" with the topical context of "rock climbing." All of these personal commonsense associations can

make the photo retrieval process more robust by allowing associated concepts to be used interchangeably in the retrieval process. Examples of this type of personal commonsense can include:

- Mark is often found in/at the user's office.
- Jane is often talked about in the context of the topic, "rock climbing".

4.1.2 Methods for Learning

Given the types of personal commonsense the system aims to learn, what are some textual learning methods that can be applied? The types of personal commonsense described above can be separated into two groups: knowledge that can be explicitly stated (e.g.: "I went to a wedding last weekend," "Here is a picture of me and my sister Mary"), verses implied associations (e.g.: Mark and "office" frequently appearing in the same context, Jane and "rock climbing" appearing in the same context). The first group is *directly learnable* and the second group *indirectly learnable*. The method for directly learnable knowledge is pattern-matching rules against WALI's event structure parse tree. Direct learning only requires one example from the text. The method used for indirectly learnable knowledge is conceptual co-occurrence through statistical methods. Indirect learning requires multiple examples from the text (possibly spread over many interactions with ARIA).

Directly Learnable

Directly learnable facts are those which can be stated rather cleanly in one sentence clause, often in syntactically and semantically simple ways. Sentences are first sent to the WALI event structure parser, and an event structure tree is produced. Pattern-

matching rules are applied both to the event structure tree, and to structures known as *thematic views,* which are concise thematic role frame snapshots of the event structure. Thematic views distill a sentence into core thematic roles such as agent, action, patient, destination, and trajectory. For example, we give as follows the event structure tree and the thematic view of the sentence "Last weekend, I went to Ken and Mary's wedding in San Francisco."

Event Structure:

```
[ASSERTION
 [TIME
  ARIA_DATESPAN{04m01d2002y-04m03d2002y}]
 [ASSERTION
  [PERSON I]
  [ACTION go
   [PATH to
     [EVENT
       [EVENT [PEOPLE [PERSON Ken] and [PERSON Mary 's]] wedding]
       [LOCATION in
        [PLACE San Francisco]]]]]]]
```

Thematic View:

```
{MAIN_ACTION: ('go', 'ACTION')
 MAIN_AGENT: (user, 'PERSON')
 DESTINATION: [('Ken and Mary's wedding in San Francisco', 'EVENT'),
               ('Ken and Mary's wedding', 'EVENT'),
               ('San Francisco', 'PLACE')]
 MAIN_DATE: 'ARIA_DATESPAN{04m01d2002y-04m03d2002y}'
}
```

The advantage of thematic views is that they provide a summary of important elements of the even structure so that pattern-matching rules can be more easily applied. In building a thematic view, a sentence with more than one set of AGENT, ACTION, PATIENT, DESTINATION, etc. requires heuristic disambiguation of the MAIN ACTION, MAIN AGENT, etc. An example of a pattern-matching rule, which applies to thematic views, is shown here, simplified:

```
IF
     View.MAIN_ACTION[0] is in go_synset and
     View.MAIN_AGENT[1] is user and
     View.DESTINATION contains x such that x[1] is 'PLACE' and
     View.MAIN_DATE is not empty
THEN
     Set <location> to be x[0]
     Set <datespan> to be pretty_print(View.MAIN_DATE)
     Learn "The user was in <location> from <datespan>[0] to
<datespan>[1]"
```

While views provide a useful summary of event structure, some pattern-matching more appropriately takes place directly against the event structure tree. Consider, for example, the sentence, "Here is a picture of me and my sister, Mary." The event structure that corresponds to this is shown below:

Event Structure:

```
[ASSERTION
 [PLACE Here]
 [STATE be
  [THING a picture
     [PROPERTY of
       [PEOPLE_IN_PHOTO
         [PERSON me] and
         [PERSON
           [PROPERTY
             [PERSON my sister] , ]
           [PERSON Mary]]]]]]]
```

This example illustrates a very simple directly learnable fact – the user has a sister named Mary. In fact, it is arguable that the use of the comma suggests the even stronger assertion that this is the user's only sister. Below is a rule that matches this very common case of a "role" (e.g. father, mother, sister, etc.) given as an appositive to a person's name.

Appositive Role Rule:

```
IF
     currentNode.label is "PERSON" and
     isName(currentNode.value) and
     currentNode.prevSisterNode is "PROPERTY" and
     currentNode.prevSisterNode.firstChild.label is "PERSON" and
```

```
     isRole(currentNode.prevSisterNode.firstChild.value)
THEN
     Set <name> to be currentNode.value
     Set <role> to be
stripMyKeyword(currentNode.prevSisterNode.firstChild.value)
     Learn "The user has a <role> named <name>"
```

Indirectly Learnable

While some kinds of personal facts are stated quite explicitly in the text, and are thus explicitly learnable, other kinds of facts are implied by the text, or stated in such a way that our semantic understanding cannot reliably recognize these facts. For example, consider the following passage, which simulates a user's input into ARIA.

> *I got in to the office at around 9, and I felt really drowsy. Mark, being the nice guy that he is, made me some coffee to cheer me up.*

In this passage, Mark is clearly someone who works with "I." (suppose the user's name is Linda). However, we cannot create pattern-matching rules to directly learn this fact. We argue the reason is that a human level understanding is needed to draw this conclusion. Computers cannot presently understand stories like this because they lack commonsense and reasoning abilities. Consider that the following commonsense is needed to know for sure that Mark is someone who works with Linda in the office:

- People generally work 9am-5pm, so the user probably did not leave the office when Mark brought her coffee. This suggests Mark gave the user coffee while they were in the office.
- The user knows Mark fairly well, specifically, she knows him to be a nice guy. It is common for people to be familiar with co-workers in their office.

- Mark must have been physically at the office with the user to notice that she is drowsy.
- The coffee was probably made at the office because many offices have coffee machines. Mark knew where the coffee machine was. He is probably familiar with the office, i.e. he works there.
- Mark knows that a good cup of coffee can help start a groggy day, so he has probably encountered the situation of being drowsy at work before. This can possibly suggest similarities between his work and the user's work.

Through commonsense reasoning, people can eliminate implausibilities that computer story understanding systems cannot. We can eliminate that Mark came from outside the office, that they were not in the office when Mark brought the user coffee, or that he did not know the user. All of commonsense leads a person to conclude that Mark is an officemate of the user's.

Without the benefit of human-level story understanding, we cannot directly learn this fact with much confidence. Instead we must rely on a statistical approach for confidence. In this approach, we measure the conceptual co-occurrence of the two concepts "Mark" and "office." The hypothesis is that semantic proximity between these concepts will be reflected in structural and syntactic proximity. In the current implementation, we do naïve counting of the number of times a co-occurrence is seen. When that number for any particular co-occurrence has exceeded a threshold, that fact is learned. More sophisticated approaches to estimating probabilities of co-occurrences include maximum likelihood (Dempster, Laird et al., 1977), Bayesian networks (Pearl, 1988), asymmetric clustering models (Jain, Dubes, 1988), among others. It is unclear how much

improvement can be garnered through a more sophisticated learning algorithm because in our domain, we face a critical data sparseness problem.

Conceptual co-occurrence has certain advantages: that it does not require rules to be crafted, and that it provides a method by which facts not directly learnable can still be learned. There are also disadvantages: that it is a shallow approach so it is more prone to learn incorrect facts, and that it is a statistical approach which requires many examples to learn a fact.

Between the deeper pattern-matching approach for directly learnable facts and the shallower conceptual co-occurrence approach for indirectly learnable facts, we are able to cover the types of personal commonsense we want to learn. In the next section, we discuss how newly learned personal commonsense can be applied.

4.2 Applying Personal Commonsense

The purpose of learning personal commonsense in ARIA is to improve future interactions with the user. The primary application of this knowledge is to improve photo retrieval by expanding annotations with semantically similar concepts (e.g. expand "my sister" with "Mary"), just as ordinary commonsense does. By ordinary commonsense, we are now referring to commonsense as defined in Chapter 3. Another potential application is bulk annotation, whereby a piece of personal commonsense can result in many photos being annotated. Other potential applications include various photo retrieval visualization methods.

Improving Retrieval

Facts in the personal commonsense repository are used, along with ordinary commonsense, to provide photo annotation expansions. Just as ordinary commonsense helps to make semantic connections from the text to annotations that reflect obvious knowledge most people share, personal commonsense makes semantic connections that reflect knowledge obvious to the user himself/herself. Another way to look at the application of personal commonsense is as a personalization for photo retrievals.

For each type of personal commonsense discussed earlier, we show how it can be used to improve retrieval.

1. **Role and Name associations.** e.g.: "The user has a sister named Mary." This allows photos annotated with "my sister" to gain the commonsense annotation "Mary," and vice versa. The user can now retrieve photos of Mary by interchangeable terms.
2. **People and places.** e.g.: "Mark is often found in/at the user's office." Associating people with a place improves retrieval because many places the user might talk about are also associated with membership in social groups. For example, "everyone from the office," or "members of my church" refer to places as well as social groups. If the user mentions the office in the story text, photos of all the people from her office can be retrieved by their individual names.
3. **People and topics.** e.g.: "Jane is often talked about in the context of the topic 'rock climbing.'" Similar to places, topics might also imply an associated social group.

Bulk Annotation

One valid criticism of ARIA is that when photos are first entered into the ARIA shoebox, they do not have annotations, and so there is no way to retrieve any of them. There clearly needs to be a method for bulk annotating photos so that all photos entered into ARIA have at least one annotation to start with. Though the bulk annotation application of personal commonsense mentioned here does not happen at the point when pictures are first entered into ARIA, and annotations only have confidence levels comparable to those of ordinary commonsense, the proposed approach still provides a way to access unannotated photos, which is much needed in ARIA.

Bulk annotation is enabled by personal commonsense associating a location with a date (e.g. "The user was in San Francisco from 1/1/2002 to 1/5/2002") or event with a date (e.g. "The user went to a wedding from 1/1/2002 to 1/5/2002"). Knowledge that the user was in San Francisco during the first week of January allows ARIA to cluster all the photos taken by the user during that time period, and bulk-annotate all of them with "San Francisco." Similarly, knowledge of events occurring within a time period allows all photos taken in that period to be annotated with the event.

Places and events are allowed to bulk-annotate photos within the prescribed time range, but it does not make sense to allow the reversed situation. That is, it does not make sense to annotate all photos having the annotation "wedding" with the date range from a personal commonsense fact.

Because of the uncertainty associated with personal commonsense, these bulk annotations are only commonsense annotations, are not heavily weighted, and do not

appear next to the photo in the ARIA shoebox. These annotations are only meant as a safety net, when no annotated photos can be retrieved.

Visualizations

Much of personal commonsense embodies relationships of the user himself/herself to places, events, time, and familiar people. This knowledge can potentially enable visualizations for information retrieval. Knowing places the user has visited might lead to a world map visualization with the user's travels charted out. Knowing when the user made trips to places and events can lead to a timeline visualization. Associations between people and the names of their relationships to the user can help build a "personal relationships" graph, of family, friends, and acquaintances. These are some of the few ways that personal commonsense gathered by ARIA can be used to visualize relationships, which can be a great way to do information retrieval.

4.3 PCSL: Personal Commonsense Learner

PCSL consists of two rule-based learners, one pattern-matching, and one statistical in nature. These learners operate on two data structures – the event structure parse tree, and the thematic view. PCSL's architecture can be grouped into two phases: preparing data structures, and rule-based learning. As shown in Figure 15, PCSL has four components, and three databases (excluding the persistent database storing the personal commonsense knowledge).

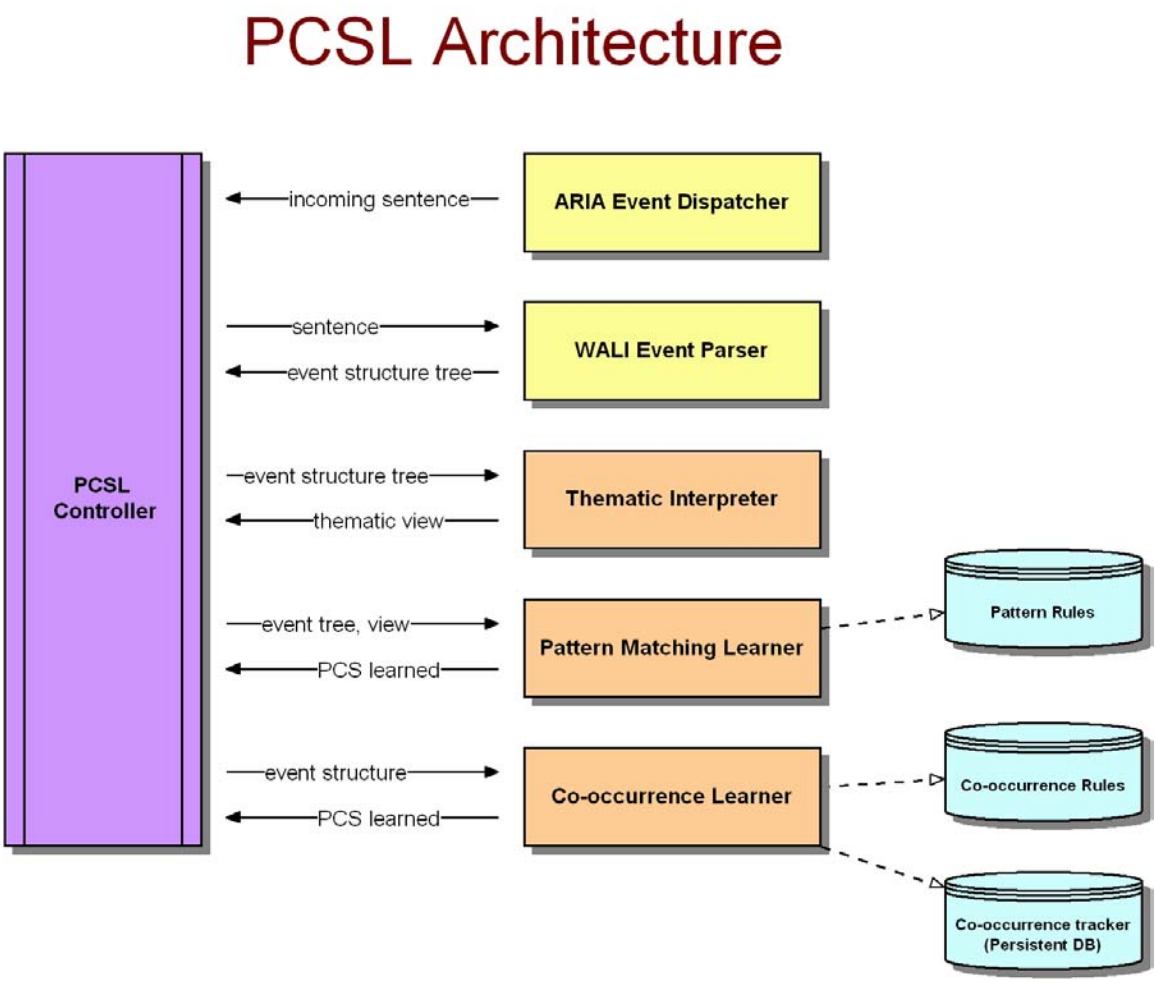

Figure 15: Overview of the PCSL Architecture. Components represented as black-boxes.

Each of the four ensuing sections presents the implementation of a component of PCSL.

4.3.1 WALI Event Parser

When the ARIA dispatcher sends a raw sentence to PCSL, it is first sent to WALI for event parsing, which was described in Chapter 2. The sentence is normalized, syntactically parsed, the resulting syntactic constituent tree is semantically transformed,

and semantic refinement makes guesses at the semantic type of unknown concepts and performs semantic highlighting. The event structure tree is needed by both the pattern-matching learner and the co-occurrence learner.

4.3.2 Thematic Interpreter

The thematic interpreter generates thematic role views, a data structure which is used by the pattern-matching learner. The thematic roles covered by the current implementation are limited to MAIN_AGENT, MAIN_ACTION, DESTINATION, AND MAIN_DATE, though other roles can easily be handled because each thematic role is discovered heuristically.

We revisit the sample event structure tree and thematic view for the input sentence, "Last weekend, I went to Ken and Mary's wedding in San Francisco."

Event Structure:

```
[ASSERTION
 [TIME
  ARIA_DATESPAN{04m01d2002y-04m03d2002y}]
 [ASSERTION
  [PERSON I]
  [ACTION go
   [PATH to
     [EVENT
       [EVENT [PEOPLE [PERSON Ken] and [PERSON Mary 's]] wedding]
       [LOCATION in
        [PLACE San Francisco]]]]]]
```

Thematic View:

```
{MAIN_ACTION: ('go', 'ACTION')
 MAIN_AGENT: (user, 'PERSON')
 DESTINATION: [('Ken and Mary's wedding in San Francisco', 'EVENT'),
               ('Ken and Mary's wedding', 'EVENT'),
               ('San Francisco', 'PLACE')]
 MAIN_DATE: 'ARIA_DATESPAN{04m01d2002y-04m03d2002y}'
}
```

The first thematic role to be filled is the MAIN_ACTION because all the other thematic roles may change depending on what is chosen. When there are multiple actions, we first make a candidate list. Applying heuristics, we score each candidate against a small set of features. The goal is to pick the ACTION that is semantically most crucial to the sentence. The features used for selecting this include rewarding infinitive verbs, rewarding verbs that have their AGENT role filled, and penalizing certain verbs that traditionally play supporting roles, such as "to be," "to want," "to need," "to do," "to like," etc. As mentioned earlier, it is a limitation of our event structure tree being tied to a syntactic alternation that makes this interpretation even necessary.

The MAIN_AGENT role is filled by the subject position of the MAIN_ACTION, unless that action is topicalized (passive voice) in the sentence. The MAIN_AGENT is semantically constrained to be a PERSON, PEOPLE or THING. Because one type of commonsense we have chosen to learn is the history of the user's travels, which emphasizes motion and movement, we have implemented the thematic role DESTINATION. This role is filled by any EVENT or PLACE structure nested in the PATH arguments to the MAIN_ACTION. In the above example, "Ken and Mary's wedding in San Francisco," "Ken and Mary's wedding," and "San Francisco" were all possible destinations because they were nested inside the PATH argument. The MAIN_DATE argument is heuristically identified as any TIME constituent which either subsumes the MAIN_ACTION constituent (e.g. "Last weekend, I went to …") or is nested in a TEMPORAL_POINT or TEMPORAL_SPAN argument to the MAIN_ACTION (e.g. "I went there over the weekend").

4.3.3 Pattern-Matching Learner

One of the two rule-based learners of PCSL, the pattern-matching learner uses its rules to recognize directly learnable sentences such as those discussed earlier in this chapter. The database of pattern rules are IF-THEN statements which operate on the event structure tree data structure and on the thematic view structure. The order in which rules fire does not matter because rules do not result in any changes to the event structure or thematic view. They only cause personal commonsense English sentences to be outputted to the Controller.

Personal commonsense is represented as simple and structured English sentences so as to conform to the formatting of ordinary commonsense sentences. By adding a mapping rule for each personal commonsense sentence template, these sentences can be compiled into concept node graphs reusing the same mechanism as used by CRIS.

4.3.4 Co-occurrence Learner

The current implementation of this learner uses naïve counting. The data sparseness problem (i.e., how many times will the user talk about Mark and the office in ARIA) draws into question the benefits of a more sophisticated statistical method. Because co-occurrences can occur across sentence boundaries, a persistent database maintains the last *n* sentences passed to PCSL.

Co-occurrence rules contain pairs of conceptual semantic categories to be considered. Currently, there are only two relationships to be learned, corresponding to four rules. They are:

- (PERSON, PLACE, 3, “PERSON is often found in/at PLACE”),

- (PEOPLE, PLACE, 3, "PEOPLE is often found in/at PLACE"),
- (PERSON, THING, 5, "PERSON is often talked about in the context of the topic, THING"),
- (PEOPLE, THING, 5, "PEOPLE is often talked about in the context of the topic, THING").

The third value in each rule is the counting threshold for how many examples must be encountered for the relationship to be learned. The fourth value in each rule is the sentence pattern for the personal commonsense fact corresponding to the rule. The first two rules correspond to learning the relationship between a person/people and a place. The last two rules correspond to learning the relationship between a person/people and a topic (e.g. "rock climbing"), which is not currently distinguished from other THINGs.

The co-occurrence algorithm extracts all the concepts from the event structure of the current sentence and the last *n* sentences, and two concepts are said to co-occur if they fall within a window size of *s* concepts. For example, given that ten concepts were extracted from the past five sentences, and given a window size of four, then every pair of concepts whose semantic categories are described by a rule and fall within a distance of four concepts from each other, will be considered a co-occurrence. The number of co-occurrences of each pair of concepts is tracked using a persistent database. When the tally exceeds the threshold indicated in a rule, the corresponding fact is outputted to the PCSL controller.

4.4 Limitations

Our approach to personal commonsense learning is limited by the semantic understanding capabilities of WALI, and by the difficulty of story understanding by computers in general.

The level of understanding in WALI was mainly intended for the purpose of automated annotation, and the design goal was to balance depth of understanding with robustness. The event structure tree produced by WALI differs from event structures produced by deeper semantic analyzers like Cyc-NL and UNITRAN. A conscious design choice was made to build the event structure on top of a syntactic constituent foundation. Such an approach inherits the robustness of syntactic parsing. The tradeoff is that the resulting event structure tree is tied to the specific alternation associated with syntax, so in PCSL, rules must consider syntax. One reason for thematic views that try to heuristically select the main action, agent, etc. of the event structure is to compensate for the syntactic dependency of the event structure tree. In Cyc-NL and UNITRAN, the semantic structure is quite independent of syntax.

Currently, the types of commonsense learnable by PCSL are restricted to simple association relations between concepts within certain semantic categories. Given something like the frame semantics of Cyc-NL, PCSL might be able to learn a much broader set of personal commonsense – facts which are specific to the semantics of a particular event, for example.

Other limitations we touched upon earlier in this chapter are those presented by computer story understanding. Relationships we dubbed “indirectly learnable” are quite

obvious to people, but difficult for computer story understanding programs lacking commonsense. Recall the example passage,

> *I got in to the office at around 9, and I felt really drowsy. Mark, being the nice guy that he is, made me some coffee to cheer me up.*

In this passage, it is clear to any human reader that Mark works in the user's office. However, a lot of commonsense is needed to be certain of this fact. Story understanding systems with commonsense are beyond today's state-of-the-art, so the method we rely on to learn such relationships is statistical, which by definition, is never right all the time.

So far in this thesis, we have talked in depth about each of the three major mechanisms of ARIA: semantic understanding for automated annotation, commonsense reasoning for robust photo retrieval, and observational learning of personal commonsense. The next chapter will attempt to put into perspective the overall approach of ARIA, and will frame our efforts at incorporating commonsense into ARIA with a discussion of other applications that have also attempted to incorporate commonsense.

Chapter 5

Related Work

The past few chapters have for the most part focused on specific subsystems in ARIA. This chapter takes us back to the big picture, hoping to offer some perspective on how approaches that have been proposed fit in with related research efforts from the literature. First, we compare ARIA to other work being done on the annotation and retrieval of images. Second, we discuss how our effort to incorporate commonsense in ARIA compares with other applications using commonsense and the OMCS corpus.

5.1 Annotation and Retrieval of Images

There are many consumer-oriented image annotation and retrieval systems in existence, and some of them are very sophisticated. The state-of-the-art in image annotation for consumer photography is probably best represented by Fotofile (Kuchinsky et. al., 1999). Fotofile has many useful tools to assist manual user annotation. Particularly useful is its bulk annotation capabilities, allowing groups of images to inherit annotations. Fotofile's structured metadata attributes, which resemble ARIA's Who, What, When, Where annotation frame representation, provides a useful and uniform way to represent an annotation through the attributes of creation date, location, subject, people, title, and description. Fotofile also proposes some automated

annotation through image recognition. Although Fotofile provides many useful tools to assist in manual annotation, no observational learning is done to automate annotations, and annotation and retrieval are not integrated with an image application such as email.

Jay Budzik and Kristian Hammond's Watson (2000) proposes using the context of user interactions to assist in image retrieval, but it does not even consider annotations.

ARIA is uniquely different from Fotofile and Watson in that it integrates annotation and retrieval, and uses observational learning from text as a way to automate image annotations and cue retrievals via real-time image recommendations. Neither of the aforementioned programs provides real-time recommendations of images or adaptively links text with images (Lieberman & Liu, 2002) through semantic connectedness. ARIA's extensive natural language understanding, incorporation of commonsense, and its ability to improve retrieval over time by learning personal commonsense further distinguish ARIA as an intelligent software agent, and not merely a software program for organizing and retrieving consumer photos.

ARIA's approach of observing what a user types and proactively retrieving suggested images on-the-fly was inspired by the Remembrance Agent (Rhodes & Starner, 1996) and Letizia (Lieberman, 1997). The Remembrance Agent observes user actions and proactively provides suggestions for related documents using the SMART information retrieval heuristic. Similarly, Letizia assists users browsing the Web by providing real-time suggestions for Web pages by observing user browsing.

5.2 Commonsense in Other Applications

A large portion of this thesis focuses on incorporating commonsense into ARIA so that ARIA does not make obvious mistakes such as missing semantic connections. The approaches explored are based mainly on analogy to other large-scale knowledge bases because there is very little precedent for commonsense in applications to follow. However, as intelligent software agents become more popular, and agents are increasingly involved in real-world tasks, we see commonsense in applications as an expanding and important research area. Furthermore, as the need for large-scale commonsense knowledge grows, commonsense corpora like Cyc and Open Mind Commonsense will also grow in importance. In this section, we present three applications from the recent literature that try to incorporate commonsense and/or the OMCS corpus in different ways to solve different problems. One is a calendar program, one is an adaptive search engine interface, and one is a story generation program.

5.2.1 In a Calendar Program

Muller's SensiCal (2000) is a calendar program that incorporates commonsense knowledge to help fill in blanks in calendar items (e.g. duration) and to provide a sanity check to the user when an action goes against commonsense. The knowledge for SensiCal comes from a commonsense knowledge and reasoning system called ThoughtTreasure (Mueller, 1998a; Mueller, 1998b), which contains 100,000 pieces of declarative and procedural commonsense. Closer to Cyc than OMCS, the knowledge in commonsense is encoded formally as predicates and arguments within a formal ontology of relations, and is not acquired from the general public as in the case of OMCS.

One of SensiCal's goals is to use commonsense knowledge about default values to help fill in the blanks in calendar items. Mueller gives an example of how SensiCal is able to automatically fill in the length of an appointment based on the nature of the appointment. For example, the calendar item "dinner w/Susanna" will result in the duration being automatically set to 2 hours, which is the typical length of a dinner according to the commonsense in ThoughtTreasure.

SensiCal's other goal is to help point out problems with the user's scheduling actions. The program organizes its commonsense about appointment scheduling into the categories of space and time, venues, activities, settings, food, and social. Mueller gives examples of commonsense, including the following:

- "A person can't be in two places at once" (Space and Time)
- "Restaurants do not generally serve dinner after 11 pm." (Venues)
- "Lunch is eaten around noon for about an hour." (Activities)
- "Vegetables are sold in grocery stores." (Settings)
- "Avoid restaurants that serve mostly food you don't eat." (Food)

Here is a common example that Mueller gives to illustrate how SensiCal might intervene: when the user tries to schedule a dinner with a vegetarian friend at a steakhouse, SensiCal prompts the user with the commonsense alert, "You want to take you vegetarian friend to a steakhouse?"

There are some similarities between SensiCal and ARIA's approach to commonsense. Like SensiCal, the commonsense selected for ARIA is based on appropriateness to the problem domain. For SensiCal, this means knowledge about space and time, venues,

activities, etc. For ARIA, knowledge about classification, space, scene composition, purpose, emotion, and causality were selected.

Another similarity is the integration of personal commonsense with ordinary commonsense. In SensiCal, commonsense such as "avoid restaurants that serve mostly food you don't eat" refers to a user profile or user model, which is filled in manually. In ARIA, personal commonsense is used by the same mechanism as ordinary commonsense to improve the retrieval process. ARIA distinguishes itself slightly yet significantly in that personal commonsense can be learned through interactions with the user.

There are also major differences in the approaches SensiCal and ARIA adopt. Commonsense for SensiCal is fairly specific to the problem domain of appointment scheduling, and therefore, may need to be crafted by hand. ARIA, on the other hand, gets its commonsense from OMCS, which is constantly growing and updating, and it uses this knowledge in a purely associative way, so knowledge does not need to be hand-crafted specifically for the application. The two programs also differ in their purpose. While the sanity check component of SensiCal uses commonsense to provide criticism, ARIA uses commonsense to provide robustness to retrieval.

5.2.2 In Search

Liu, Lieberman and Selker's Goal-Oriented Search Engine (GOOSE) (2002) uses commonsense to help web novices formulate good search queries. They argue that novice search engine users have trouble formulating effective search queries because they more naturally express search goals (e.g. "I want to find people who like old movies") rather than topic keywords. In addition, they may not have the expertise necessary to

formulate good searches. Commonsense in GOOSE is meant to bridge the gap between a user's goal and a search engine's requirements by assuming some responsibility for reasoning on behalf of the user. Ordinary commonsense is also integrated with application-level commonsense, which reflects search expertise.

Like ARIA, GOOSE uses the OMCS corpus, but not merely in an associative capacity. Commonsense knowledge, classified into knowledge sub-domains to be more useful, is guided by reasoning templates to perform first-order inference from the search goal to a search query. For example, given the search goal, "I want help in solving this problem: my golden retriever has a cough," GOOSE will first parse the goal into one of many semantic frames as follows.

```
Problem Attribute:      [cough]
Problem Object:         [golden retriever]
```

A classifier selects an appropriate commonsense knowledge domain to reason over (the argument for domains is the exponential search space of first-order inference). And the following inference chain is produced.

1. A golden retriever is a kind of dog.
2. A dog may be a kind of pet.
3. Something that coughs indicates it is sick.
4. Veterinarians can solve problems with pets that are sick.
5. Veterinarians are locally located.

This results in the final search query: "Veterinarians, Cambridge, MA".

Since first-order inference is performed and the knowledge in OMCS may be sparse over some topics, it is possible for many inference chains to be unsuccessful. The evaluation presented by GOOSE proved this point. It showed that while successful inferences by GOOSE helped to improve search results, inferences were brittle and failed to be successful in a large number of cases. However, the authors argue that GOOSE can

be useful because like ARIA, it is a fail-soft application. When added to a search engine, GOOSE will produce a better search result occasionally, and the remainder of the time its failures will not prevent the search on the user's original query from going through.

5.2.3 In Story Generation

Liu and Singh's story generation agent, MAKEBELIEVE (2002), demonstrated an interesting use of the OMCS corpus. Rather than applying commonsense knowledge in the traditional sense, i.e., preventing obvious mistakes, or doing reasoning, MAKEBELIEVE extracted and used a subset of the knowledge in OMCS about causality for story generation. 9,000 sentences like "a consequence of drinking alcohol is intoxication" were extracted from OMCS and parsed into trans-frames (before and after frames), a knowledge representation advocated by Minsky in his Society of Mind (1986). Then, by chaining these trans-frames and adding a protagonist chosen by the user, MAKEBELIEVE is able to generate short fictional stories of 5 to 15 lines. Though no narrative goals were encoded in the generation process, users in the evaluation claimed that stories had plot devices like comedy, irony, etc. This is an example of how people tend to attribute intentionality (Dennett, 1989) to anything that might seem to be intelligent.

Like ARIA, MAKEBELIEVE validates the diversity of the relationships captured by the knowledge in OMCS. It also suggests that a large-scale commonsense knowledge base might have other applications than merely preventing obvious mistakes as in ARIA, or providing critique, as in SensiCal.

This chapter has hopefully helped to place ARIA and the commonsense capacities of ARIA into the context of other research efforts in these areas. In the next chapter, we bring the big picture back to ARIA to discuss the contributions made by this thesis work and to suggest future work needed.

Chapter 6

Conclusions

In this thesis, we investigated three broad properties of intelligent software agents – communication through and understanding human language; exercising some commonsense to prevent obvious mistakes; and learning from past user interactions to improve future interactions. The ARIA Photo Agent provided an application platform from which approaches to these properties were tested, and what resulted was an intelligent software agent that can automatically annotate photos by understanding the user's English text; robustly retrieve annotated photos by incorporating commonsense; and learn personal commonsense of the user and use that knowledge to improve future retrieval by knowledge specific to the user.

Understanding human language is an integral goal of ARIA, which aims to automate annotation of images based on their use in an email text. In order to learn annotations, ARIA must understand the semantics of the text well enough to extract relevant annotations from descriptions. However, robustness is also a concern, and automated annotation cannot be whimsical, because photos may be used only once before they are next needed much later. The approach taken in this thesis produced WALI, which represents a hybrid approach between shallow, robust syntactic analyzers, and deeper, but more brittle semantic analyzers. Building Jackendoff-style event structures on top of

syntactic constituents allowed our semantic parse structure to inherit the robustness of the syntactic parser, while event structure information and world semantic categorization of concepts allowed our parser to output a much richer representation than that of syntactic parsers. The extra structural information given by our representation helped the personal commonsense learner, while the extra semantic information made automated annotation extraction from text fairly straightforward. With robustness came a tradeoff of understanding, however. Since event structures were built on top of syntactic constituents, the resulting semantic event structure was not free from syntax; instead, event structure was tied to syntactic phenomena like specific surface realizations and verb alternations. Although this did not render WALI's parse to be useless, it made understanding the event structure tree with rules (as in PCSL) to be much more difficult, because multiple rules had to be created to handle different syntactic alternations of the same underlying event structure.

Commonsense reasoning is integral to another one of ARIA's goals – robust photo retrieval. Without commonsense, the retrieval of annotated photos often misses obvious semantic connections such as the connection between "bride" and "wedding." In this thesis, CRIS provides an associative commonsense reasoning mechanism to help improve retrieval robustness. CRIS also proposes special case reasoning and cross-annotation mechanisms for temporal and geospatial reasoning. By compiling knowledge from OMCS, a publicly acquired corpus of commonsense, into an efficient semantic resource, CRIS is able to incorporate large-scale commonsense knowledge on the order of 80,000 facts into real-time photo suggestion. We showed that even though a dirty corpus like OMCS has an ambiguous representation, and is not filtered of noise, we were still able to

readily integrate the knowledge to improve ARIA through associative reasoning. While associative reasoning is generally weaker than inference, it is perfectly suited to making semantic connections between text concepts by enabling commonsense expansions of annotations. In addition, this shallow reasoning mechanism is not prone to the problems associated with first-order inference, such as exponential search space, and requiring knowledge of inference patterns. When compared with other applications that have used commonsense, such as in search (GOOSE), in storytelling (MAKEBELIEVE), or in a calendar program (SensiCal), the 80,000 facts used by ARIA is the largest knowledge set, and uses the most diverse set of commonsense relationships, all taken from the generic (not hand-crafted, not custom made for the application) OMCS corpus.

Learning is also useful to ARIA. Learning personal commonsense about the user, such as the user's personal relationships with people, and places the user has visited, ARIA can later use this knowledge to make semantic connections which are obvious to the user, such as the connection between the concepts "Mary" and "my sister." Learned knowledge also has potential to help solve the bulk annotation problem for ARIA, and may also be used to create visualizations for photo retrieval. Our approach to learning invokes two methods to learn both explicitly stated facts (directly learnable) as well as implied facts (indirectly learnable). The event structure parse generated by WALI, along with thematic views, allowed for pattern-matching learning of certain types of commonsense. The shallow back-off statistical learner of conceptual co-occurrence complemented well the abilities and limitations of the pattern-matching learner.

The next section in this chapter summarizes the contributions made by this thesis to research in the areas of semantic parsing, commonsense for applications, user modeling,

and software agents in general. We conclude with a discussion of future directions for ARIA.

6.1 Summary of Contributions

We hope that the approaches, methods, and techniques explored by this thesis will help to further the understanding of solutions and limitations in the research areas of semantic understanding, commonsense, user modeling, and software agents.

Semantic Understanding

We have proposed a semantic understanding mechanism called WALI for the class of problem domains that require a balance between robustness and depth of understanding, such as ARIA. Syntactic parsing is robust, but provides no semantic information, so it is often insufficient for tasks that require a semantic understanding of text, such as automated annotation of photos using text descriptions. Semantic parsers like Cyc-NL and UNITRAN provide a deep level of understanding of the underlying semantics of text, and the outputted representations are freed completely from syntactic structure. Unfortunately, such semantic parsers are brittle because they rely on a slew of resources such as verb-argument structure, which are incomplete at best. In our hybrid approach, "understanding tools" such as semantic typing of concepts and Jackendoff-style event structure are built by applying semantic understanding agents (UAs) and a transformational grammar to a foundation of syntactic constituents. Because the constituent structure derives from syntactic parsing, the robustness of our structure is comparable. At the same time, the event structure and semantic typing of concepts provides a deeper level of semantic understanding than syntactic parsing can claim. We

believe that our semantically informed syntactic parsing provides the right balance between robustness and depth and can be useful in our problem domains that have such criteria, such as in information extraction, shallow question answering, and command and control.

Another contribution is our world-aware understanding agents. Traditionally in natural language processing, individual words may be tagged with parts-of-speech, and some types of concepts such as email addresses and name are also tagged. This is called named-entity recognition. However, we are not aware of any other efforts to assign types to world-semantic categories, that is to say, a broad set of concepts common in the everyday world. In WALI, these types included TIME, PERSON, PLACE, EVENT, THING, and EMOTION. The challenge is that there exists a very large number of items of each type in the real world, and this has likely limited lexicon building efforts. Our approach is scalable and extensible because we automatically mine this knowledge out of a freely available, publicly acquired, and ever-growing corpus of commonsense called OMCS. As the quantity and quality of knowledge in OMCS grows, so will the world-semantic recognition abilities of WALI. World-aware understanding agents can be useful to any program that requires some understanding of the everyday world, such as in story understanding, or command and control.

Commonsense for Applications

We have demonstrated how commonsense coming from a generic commonsense knowledge source can be used to make an application like ARIA more robust. We have shown that commonsense knowledge can be just as useful in shallow associative reasoning, as in traditional ideas of reasoning such as first-order inference. Furthermore,

we demonstrated how noise and ambiguity in OMCS could be heuristically managed as that knowledge is compiled into a semantic resource. By using an efficient method like spreading activation rather than inefficient inference chains, 80,000 pieces of commonsense could be used to improve ARIA in real-time. We hope that the successful incorporation of commonsense in ARIA will lead to many more applications with commonsense. We also hope that ARIA's success will help to bring to light the value of a large-scale generic commonsense knowledge base like OMCS, and also to validate its public acquisition approach to knowledge gathering.

User Modeling

Our approach to user modeling is different from traditional notions of user modeling as building user profiles. Instead of modeling the externalities of a user, such as their demographics, products they bought, etc., we model what they know, namely, what sort of knowledge is obvious to them. We believe that ordinary commonsense knowledge provides the basis for a generic user model, because it is knowledge that most people obviously know. Personal commonsense knowledge becomes a way to specialize that user model to a user. By personal commonsense, we mean both knowledge about the user (e.g. "Mary is my sister"), as well as knowledge the user knows to be obvious (e.g. "Mark works in my office"). By combining pattern-matching learning and co-occurrence learning with our semantic understanding mechanism, we are able to learn a broader set of personal commonsense relationships.

Software Agents

This thesis has demonstrated how semantic understanding, commonsense, and learning have helped to make ARIA behave more intelligently by understanding language, making fewer obvious mistakes, and growing its capabilities through continued use. We hope that future researchers will consider the value of these three characteristics when building intelligent software agents.

6.2 Future Work

In future work on ARIA, we plan to improve the scalability of the photo shoebox, add bulk annotation capabilities, investigate speech annotation, incorporate OMCS version 2 as a knowledge source, and conduct a formal evaluation of the system. Going beyond ARIA, we would like to apply ARIA's approach to other domains. We would also like to use the semantic understanding approach of WALI for knowledge extraction from the World Wide Web.

Scalability of ARIA's shoebox

As a user's photo collection grows in ARIA, there will need to be a way to group together similar photos in the shoebox so that the size of the shoebox stays reasonable. One way to accomplish this is by grouping together photos by time, or by event. It may even be desirable to annotate these groups of pictures, and for pictures within each group to annotate the group's annotations. For example, all the photos of Ken and Mary's wedding can be grouped together and annotated with "Ken and Mary's wedding." Then each individual photo in that collection can have more specific annotations. Photo suggestions can instead retrieve these photo clusters, and if a user chooses to expand a

cluster, individual photos can be suggested. Adding photo clusters will allow ARIA's shoebox to scale well to large photo collections.

Bulk Annotation

As we mentioned in Chapter 4, one of the most valid criticisms of the current ARIA system is that when photos are first entered into ARIA, they have no annotations by which they can be retrieved. To address this, we propose adding a bulk annotation mechanism to annotate all incoming photos by either the event or place of the photos so that each photo has at least one handle by which it can be retrieved.

Speech Annotation

Speech annotation is another capability we would like to investigate. Just as text descriptions from an email can be used to annotate photos, speech descriptions of photos presented in a slideshow should also be useful to learning annotations. The challenges presented by speech are the accuracy problems posed by most state-of-the-art speech recognition systems like IBM ViaVoice and L&H Dragon Naturally Speaking. Missing words or incorrectly transcribed words can propagate failure to parsing, which in turn will prevent annotations from being learned correctly. However, speech retrieval is a related mechanism that has more immediate potential, because it only requires phrases in speech to be spotted and matched against annotations. If speech retrieval is made contextual, we can imagine it being used in settings like meetings. As a presenter talks about a particular chart, that graphic can be quickly retrieved.

OMCS Version 2

The first version of OMCS, used as the commonsense knowledge source for ARIA, has problems with noise and ambiguity, because of blanks in the English sentence templates are under-constrained, and word senses are not disambiguated. The next version of OMCS, based on commonsense XML templates, will be more structured, will have disambiguated word senses, and will provide a cleaner source of knowledge for the commonsense reasoning mechanism in ARIA. We hope to migrate to OMCS version 2 when it becomes available.

Formal Evaluation

An extensive evaluation of a keyword-driven version of ARIA was made at Kodak Labs. The results of that user test validated the approach to integrating annotation and retrieval with a user task like emailing and posting web pages. However, the work of this thesis on semantic understanding, commonsense reasoning and learning has not yet been formally evaluated by user tests. We hope that such an evaluation will focus on the effectiveness of our semantic understanding approach against the keyword-driven baseline approach, and to measure the effectiveness of commonsense and learned personal commonsense in improving the robustness of photo retrieval.

Adapting ARIA to other domains

ARIA's approach to annotation and retrieval through contextual information can be adapted to other domains. Given news photos embedded in newspaper articles, or photos embedded in web pages, ARIA's semantic understanding mechanism might be used to annotate photos embedded in a variety of text sources outside email texts. Objects being

annotated are not restricted to photos. We can imagine the annotation and retrieval of video clips to assist a filmmaker, or the annotation and retrieval of medical records to assist health care professionals.

WALI for Knowledge Extraction Over the Web

A much loftier goal for future work is to apply the semantic understanding approach of WALI to knowledge extraction from very large corpora like the World Wide Web. The World Wide Web corpus is one of the largest available, and is very information rich; however, broad knowledge extraction has been difficult. Deep semantic parsers representing knowledge as logic (Cyc-NL) or primitive actions (UNITRAN) have been too brittle to apply to the web. On the other hand, keyword spotting and collocation techniques have achieved too shallow of an understanding to allow the extraction of interesting and useful knowledge from the web, such as answers to questions. If WALI can have more success at broad knowledge extraction than semantic parsing or keyword spotting, it would provide important validation for our semantic understanding approach. It might also bring us closer to the ultimate exploitation of the wealth of knowledge of the World Wide Web, which is still waiting to be extracted.

Chapter 7

References

Baker, C.F., Fillmore, C.J., and Lowe, J.B. (1998). The Berkeley FrameNet project. In *Proceedings of the COLING-ACL*, Montreal, Canada.

Brill, E. (1995). *Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging*. Computational Linguistics, 21 (4), 543--565.

Budzik, J. and Hammond, K. J. (2000). User Interactions with Everyday Applications as Context for Just-in-Time Information Access. *Proceedings of the ACM Conf. Intelligent User Interfaces (IUI 2000)*, pp.44-51, ACM Press, New York.

Buitelaar, P. (1998). *CoreLex: Systematic Polysemy and Underspecification.* Ph.D. thesis, Computer Science Department, Brandeis University, Feb.

Burns, K. J., & Davis, A. R. (1999). Building and maintaining a semantically adequate lexicon using Cyc. In Viegas, Evelyne. (Ed.), *Breadth and depth of semantic lexicons*. Dordrecht: Kluwer.

Chomsky, N., (1981). *Government and Binding*, Foris, Dordrecht.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B, 39:1#38.

Dennett, D.C. (1989). *The Intentional Stance.* MIT Press, Cambridge, MA.

Dorr, B. J. (1992). The use of lexical semantics in interlingual machine translation. *Journal of Machine Translation*, *7*(3), 135-193.

Dowty, D., Wall, R., and Peters, S. (1981) *Introduction to Montague Semantics*. Reidel Publishing Company.

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Fillmore, C. J. (1968). The case for case; in Bach and Harms (Ed.): *Universals in Linguistic Theory* (pp. 1-88), Holt, Rinehart, and Winston, New York.

Grishman, R., Macleod, C., & Meyers, A. (1994). COMLEX Syntax: Building a computational lexicon. In *Proceedings of the 15th COLING*. Kyoto, Japan.

Hale, K., and Keyser, J. (1986a). *Some Transitivity Alternations in English*, Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA, Lexicon Project Working Papers #7.

Hale, K., and Keyser, J. (1986b). *A View from the Middle*, Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA, Lexicon Project Working Papers #10

Hale, K., and Keyser, J. (1989). *On Some Syntactic Rules in the Lexicon*, Center for Cognitive Science, Massachusetts Institute of Technology, Cambridge, MA, manuscript.

Hale, K., Laughren, M. (1983). *The Structure of Verbal Entries: Preface to Dictionary Entries of Verbs*, Massachusetts Institute of Technology, Cambridge, MA, Warlpiri Lexicon Project.

Jackendoff, R. (1983). *Semantics and Cognition.* Cambridge, MA: MIT Press.

Jackendoff, R. (1990). *Semantic structures.* Cambridge, MA: MIT Press.

Jackendoff, R. (1993). X-bar Semantics. In Pustejovsky (ed.), *Semantics and the Lexicon,* Kluwer, Dordrecht, pp. 15-26.

Jain, A. and Dubes, R., (1988). *Algorithms for Clustering Data.* Prentice Hall, New Jersey.

Karttunen, L., Kaplan, R.M., and Zaenen, A. (1992). Two-level Morphology with Composition. In *Proceedings of COLING-92*, 141-- 148.

Klink, S., (2001). Query reformulation with collaborative concept-based expansion. *Proceedings of the First International Workshop on Web Document Analysis*, Seattle, WA.

Kuchinsky, A., Pering, C., Creech, M. L., Freeze, D., Serra, B., and Gwizdka, J. (1999). FotoFile: a consumer multimedia organization and retrieval system. *Proceedings of the ACM Conference on Human-Computer Interface*, (CHI-99) pp. 496 – 503, Pittsburgh.

Lenat, D. (1998). *The dimensions of context-space*, Cycorp technical report, www.cyc.com.

Levin, B. (1993). *English verb classes and alternations*. Chicago: University of Chicago Press.

Lieberman, H. and Liu, H. (2002). Adaptive Linking between Text and Photos Using Common Sense Reasoning. *In Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, (AH2002) Malaga, Spain.

Lieberman, H., Rosenzweig E., Singh, P., (2001). Aria: An Agent For Annotating And Retrieving Images, *IEEE Computer*, July 2001, pp. 57-61.

Lin, D., (1998). Using collocation statistics in information extraction. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, San Francisco, CA.

Liu, H., Lieberman, H. (2002). Robust photo retrieval using world semantics. *Proceedings of the 3rd International Conference on Language Resources And Evaluation Workshop: Using Semantics for Information Retrieval and Filtering (LREC2002),* Las Palmas, Canary Islands.

Liu, H., Lieberman, H., Selker, T., (2002). GOOSE: A Goal-Oriented Search Engine With Commonsense. *Proceedings of the 2002 International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Malaga, Spain.

Liu, H., Singh, P., (2002). MAKEBELIEVE: Using Commonsense to Generate Stories. *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-02)* -- Student Abstracts and Posters Program. Seattle, WA.

Linguistic Data Consortium, *Wall Street Journal Corpus*, Release 0, Philadelphia, PA.

Marcus, M.P., Santorini, B., and Marcinkiewicz, M.A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank, in Computational Linguistics, Volume 19, Number 2, pp. 313--330 (Special Issue on Using Large Corpora).

Minsky, M. (1986) *The Society of Mind.* New York: Simon and Schuster.

Mitkov, R., (1998). Robust Pronoun Resolution with Limited Knowledge. In *Proceedings of the 36th Annual Meeting of the Association of Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL98)*, 869-875.

Mueller, E.T. (1998a). *Natural language processing with ThoughtTreasure*. New York: Signiform.

Mueller, E.T. (1998b). *ThoughtTreasure: A natural language/commonsense platform* (Online). Available: http://www.signiform.com/tt/htm/overview.htm

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan-Kauffman.

Peat, H. J. and Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the ASIS*, 42(5), 378-383.

Ramshaw, L.A. & Marcus, M.P., (1995). Text Chunking using Transformation-Based Learning. *ACL Third Workshop on Very Large Corpora*, pp. 82-94. Available: http://citeseer.nj.nec.com/ramshaw95text.html

Richardson, S.D., Dolan, W.B., Vanderwende, L., (1998). MindNet: Acquiring and Structuring Semantic Information from Text. *Proceedings of the joint ACL and COLING conference*, 1098-1102, Montreal.

Ritchie, G.D., Russell, G.J., Black, A.W., and Pulman, S.G. (1992). *Computational Morphology: Practical Mechanisms for the English Lexicon*. MIT Press, Cambridge, Mass.

Salton, G., and Buckley, C., (1988). On the Use of Spreading Activation Methods in Automatic Information Retrieval. *Proceedings of the 11th Ann. Int. ACM SIGIR Conf. on R&D in Information Retrieval* (ACM), 147-160.

Sanfilippo, A. (1990). *A morphological analyzer for English and Italian*. In Sanfilippo, A. eds. The (Other) Cambridge Acquilex Papers. Cambridge University, Computer Laboratory, TR-92-253: 49--68.

Singh, P., (2002). The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. Palo Alto, CA, AAAI.

Sleator, D., & Temperley, D. (1993). Parsing English with a link grammar. In *Third International Workshop on Parsing Technologies*.

Voorhees, E., (1994). Query expansion using lexical-semantic relations. *Proceedings of ACM SIGIR Intl. Conf. on Research and Development in Information Retrieval*, pages 61—69.

XTAG Research Group, (1999). *A lexicalized tree adjoining grammar for English* (Online). Philadelphia, PA: University of Pennsylvania.

Xu, J., and Croft, W.B., (1996). Query Expansion Using Local and Global Document Analysis. Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 4—11.

Zubizarreta, M.L. (1982). *On the Relationship of the Lexicon to Syntax*, Ph.D. thesis, Massachusetts Institute of Technology.

Zubizarreta, M.L. (1987). *Levels of Representation in the Lexicon and in the Syntax*, Foris Publications, Dordrecht, Holland / Cinnaminson, USA.